MIXED-INITIATIVE MULTIMEDIA FOR MOBILE DEVICES: DESIGN OF A SEMANTICALLY RELEVANT LOW LATENCY SYSTEM FOR NEWS VIDEO RECOMMENDATIONS

A Dissertation
Presented to
The Academic Faculty

by

Jeannie Su Ann Lee

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
in
Electrical and Computer Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
August 2010

MIXED-INITIATIVE MULTIMEDIA FOR MOBILE DEVICES: DESIGN OF A SEMANTICALLY RELEVANT LOW LATENCY SYSTEM FOR NEWS VIDEO RECOMMENDATIONS

Approved by:

Dr. Nikil Jayant, Advisor
Professor, School of ECE
Georgia Institute of Technology

Dr. Henry Owen
Professor, School of ECE
Georgia Institute of Technology

Dr. Vijay Madisetti
Assoc. Professor, School of ECE
Georgia Institute of Technology

Dr. Janet Murray
Professor, School of LCC
Georgia Institute of Technology

Dr. Ghassan Al-Regib
Assoc. Professor, School of ECE
Georgia Institute of Technology

Date Approved: June 29, 2010

ACKNOWLEDGEMENTS

This dissertation would not have been possible without the support and encouragement of many people.

I am very much grateful to my advisor, Professor Nikil Jayant for his patience, guidance, and understanding over the years. He allowed flexibility in the choice of topic, and valuable input at every stage. I have learnt many things from him. With his help, I was able to complete my graduate studies without worrying about financial support.

My committee did undergo some changes, but this has provided more opportunity to receive valuable feedback from the ECE faculty. The proposal committee was chaired by Dr Joel Jackson, and the defence committee includes Professor Vijay Madisetti, Professor Ghassan Al-Regib, Professor Janet Murray, and Professor Henry Owen. I am thankful for all their feedback, questions, and input on research directions needed to complete the dissertation. They have also been very understanding of my situation and agreed to meet at short notice, despite their busy schedules.

I also thank my former advisor Professor Ramesh Jain, for many inspiring and philosophical discussions that helped shape my understanding of search and multimedia information retrieval, and helping me to join the PhD program. He made sure all his students were taken care of and supported, despite fighting cancer.

Many thanks also goes to Rhandeev Singh and Michael Morrissey of Google for the contribution of an Android Dev Phone 1 (HTC Dream) in support of the research demo and user experiments.

The GCATT staff consisting of Barbara, Rex, Tina and JoAnna have been extremely helpful and efficient in the handling of all administrative details. I appreciate

all their help rendered, allowing me to focus on the research work.

I am most indebted to Yu-Xi Lim, who has always been by my side and a source of encouragement and emotional support through the years and various ups and downs. Despite everything, he has always been there for me, unconditionally.

Finally, words cannot express my gratitude for my family. My parents Richard and Lily and my brother Danny, they have always been there for me and believed in me, providing unconditional love and the resources that enabled me to come this far.

I have spent the better part of a decade at Georgia Tech and have come to know many friends and colleagues, too many to list here. This includes the Singaporeans at Georgia Tech, the "CS Gang", and present and past members of the Multimedia Communications Lab. The grad school experience would not have been the same without them. I wish them all the best in their future endeavours and to always keep in touch.

This dissertation almost didn't happen. I was wondering if I would ever see this day as I was lying in hospital after getting hit by a vehicle. I am grateful to everyone who rendered assistance during that dark period, and I hope there will be an opportunity to return that favour sometime in the future.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ASR automatic speech recognition.

CBIR content based image retrieval.

FTP File Transfer Protocol.

GPS Global Positioning System.

GUI graphical user interface.

HTTP Hypertext Transfer Protocol.

IR information retrieval.

JSON JavaScript Object Notation.

MIR multimedia information retrieval.

OCR optical character recognition.

PDA portable digital assistant.

PDV packet delay variation. The difference in end-to-end delay between packets, also less precisely known as delay jitter.

RF relevance feedback. A technique that integrates user interaction and continuous repeated feedback from the user towards learning more about the user's query.

RTP Real-time Transport Protocol.

RTSP Real Time Streaming Protocol.

RTT round trip time. The time required to send a signal in both directions over a particular communication link, and the soonest time possible to receive an acknowledgement of a message.

TCP Transmission Control Protocol.

tf-idf term frequency inverse document frequency. A statistical measure that evaluates how important a word is relative to a document in a collection or corpus.

UDP User Datagram Protocol.

UI user interface.

SUMMARY

The increasing ubiquity of networked mobile devices such as cell phones and PDAs has created new opportunities for the transmission and display of multimedia content. However, any mobile device has inherent resource constraints: low network bandwidth, small screen sizes, limited input methods, and low commitment viewing. Mobile systems that provide information display and access thus need to mitigate these various constraints. Despite progress in information retrieval and content recommendation, there has been less focus on issues arising from a network-oriented and mobile perspective.

This dissertation investigates a coordinated design approach to networked multimedia on mobile devices, and considers the abovementioned system perspectives. Within the context of accessing news video on mobile devices, the goal is to provide a cognitively palatable stream of videos and a seamless, low-latency user experience. Mixed-initiative—a method whereby intelligent services and users collaborate efficiently to achieve the user's goals, is the cornerstone of the system design and integrates user relevance feedback with a content recommendation engine and a content- and network-aware video buffer prefetching technique. These various components have otherwise been considered independently in other prior system designs.

To overcome limited interactivity, a mixed-initiative user interface was used to present a sequence of news video clips to the user, along with operations to vote-up or vote-down a video to indicate its relevance. On-screen gesture equivalents of these operations were also implemented to reduce user interface elements occupying the screen. Semantic relevancy was then improved by extracting and indexing the content

of each video clip as text features, and using a Naïve Bayesian content recommendation strategy that harnessed the user relevance feedback to tailor the subsequent video recommendations. With the system's knowledge of relevant videos, a content-aware video buffer prefetching scheme was then integrated, using the abovementioned feedback to lower the user perceived latency on the client-end.

As an information retrieval system consists of many interacting components, a client-server video streaming model is first developed for clarity and simplicity. Using a CNN news video clip database, experiments were then conducted using this model to simulate user scenarios. As the aim of improving semantic relevancy sometimes opposes user interface tools for interactivity and user perceived latency, a quantitative evaluation was done to observe the tradeoffs between bandwidth, semantic relevance, and user perceived latency. Performance tradeoffs involving semantic relevancy and user perceived latency were then predicted.

In addition, complementary human user subjective tests are conducted with actual mobile phone hardware running on the Google Android platform. These experiments suggest that a mixed-initiative approach is helpful for recommending news video content on a mobile device for overcoming the mobile limitations of user interface tools for interactivity and client-end perceived latency. Users desired interactivity and responsiveness while viewing videos, and were willing to sacrifice some content relevancy in order gain lower perceived latency.

Recommended future work includes expanding the content recommendation to incorporate viewing data from a large population, and the creation of a global hybrid content-based and collaborative filtering algorithm for better results. Also, based on existing user behaviour, users were reluctant to provide more input than necessary. Additional user experiments can be designed to quantify user attention and interest during video watching on a mobile device, and for better definition and incorporation of implicit user feedback.

CHAPTER 1

INTRODUCTION

With the emergence of portable networked media devices such as iPods, portable digital assistants (PDAs) and multimedia cell phones, consumers demand access to multimedia content on-the-go. Hardware advances, increasing bandwidth and lower data costs has also enabled streaming video content on mobile devices to become increasingly ubiquitous.

However, a mobile platform always poses unique challenges due to its intrinsic constraints: For any given level of technology and cost, mobile devices may improve in absolute capabilities but will always remain resource-poor relative to their desktop counterparts due to their portability and mobility requirements [4]. These problems include low processor speed and limited network bandwidth, reduced resolution, small screen size, and limited input methods. As fidelity is traded off for mobility and accessibility, these issues change user browsing styles and inevitably lead to users having short attention spans, low commitment and extremely short interaction intervals [5].

Additionally, a mobile device also has different affordances [6]. The latest devices have features such as touch screens, accelerometers, Global Positioning System (GPS) and light detectors that relay information about the location, physical environment and accept tactile input. This allows the creation of more sophisticated interaction methods such as direct manipulation and gestures to mitigate the abovementioned issues with mobile video viewing. Direct operations on the screen or on camera also reduces the number of displayed user interface (UI) elements and frees screen estate.

Therefore, a rethinking of traditional approaches to information display, access and delivery is required, and complicates the design of such mobile systems and applications. Multimedia information retrieval (MIR) and delivery already inherently draws from a large multidisciplinary research field, since a wide range of issues related

to several research areas must be addressed in the design of such systems. Examples of these are how to extract semantics and represent multimedia data, indexing, archiving, searching and displaying such data, and how to deliver them efficiently through the network. Each area has addressed their respective problems independently in the context of a standalone desktop workstation and not from the perspective of a networked mobile device.

Existing literature has prescribed some general guidelines [5, 7, 8] for multimedia display on a mobile device: User input should be minimised and simplified where possible, without requiring intense and detailed inspection of the screen. This requires careful design of the mobile UI to maximise screen space, such as converting spatial information into a temporal format. Information should also be filtered so that only the most important items are readily accessed, rather than a display crammed with all information. This leans towards more pre-processing and effective organisation and search using information retrieval (IR) techniques to proactively recommend items, and consequently demand less interaction on the user's part. High performance algorithms will also be necessary for responding to a query in an acceptable time period and achieving effective use of network bandwidth [9]. Especially for mobile devices, low network bandwidth is a bottleneck to interactivity due to waiting times for media download, and user perceived waiting time will be high at lower bandwidths.

Techniques from each area can be examined in the context of a mobile environment and will provide combined solutions to mitigate these various mobile constraints. Coordinated design of the content retrieval and delivery system will then strike a balance between competing concerns and create a strategy to address the problem of limited resources and harness a mobile device's affordances. Recent work has attempted to create compelling multimedia information retrieval (MIR) systems for the mobile environment to address these issues [7, 10, 11].

This dissertation presents a combined system design approach to networked multimedia on mobile devices, and considers the system perspectives of the UI, content and network, focusing on semantics, content relevancy and user behaviour. Within the context of accessing news video on mobile devices, the goal is to provide a cognitively palatable stream of videos and a seamless, low latency user experience. Mixed-initiative—a method whereby intelligent services and users collaborate efficiently to achieve user goals [12, 13], forms the basis of the design and integrates user relevance feedback (RF) with a content recommendation engine, and a content- and network-aware video buffer prefetching technique.

The adaptive mixed-initiative interface uses a multi-modal "interactive TV-channel" metaphor [14], where the user is presented with a sequence of news video clips and operations to vote-up, vote-down or skip a video to indicate its relevance. Each operation has its corresponding up/down and horizontal on-screen finger gesture [15], and buttons on-screen. The user's RF is then fed to a Naïve Bayesian classifier used to determine subsequent video recommendations in the sequence. Apart from this semi-automatic user-machine synergistic feedback mode, the system can also function in an automatic mode, where videos are "pushed" to device and flip automatically from one video clip to the next, or a manual menu selection mode where videos are selected to be "pulled" in for view. Shaking the device will reset the video sequence and votes. Simultaneously, the feedback also assists client-side content-based prefetching of the video prefixes to decrease the user's perceived waiting time for video buffering [16, 17].

To simplify and bring focus to the main parameters, a client-server streaming video delivery model of information retrieval is first developed. A CNN news video clip database is then assembled and used to perform user scenario simulations to observe the benefits of more relevant content due to content recommendation, and user

perceived latency improvement due to prefetching, compared to a random presentation of video clips. Different prefetch strategies can be employed depending on the usersŕequirements and tolerance for content relevance versus wait time. The goals of content relevance and latency often oppose each other, thus their tradeoffs are examined.

Finally, the application is implemented on the HTC Dream (T-Mobile G1) on the Google Android platform, making use of the touch screen and in-device accelerometer. A human user experiment was conducted, evaluating the real world usage of such a prototype system comparing different conditions of content relevance and user perceived latency. Based on the user experiment results, there is a significant user preference for more relevant content, and that a mixed-initiative interface is useful for news video viewing on a mobile device. Users also desired lower perceived latency when viewing videos and were willing to sacrifice some content relevancy in order to achieve this.

1.1 Research Scope

Multimedia information retrieval (MIR) is inherently multi-disciplinary, drawing expertise from several research areas in the process of building the components of such a system, such as indexing, representation, searching and network delivery. The main components and processes in a MIR system are described in detail in Section 2.1.

As highlighted in blue in Figure 1, this dissertation focuses on the online and user-facing components of an IR system and the coordinated design between these components.

Specifically, the components are the mobile user interface (UI) (Chapter 3), the improvement of content and semantics relevancy through the development of a content recommendation algorithm (Chapter 5), the reduction of client-side user perceived latency via network prefetching of the video prefixes (Chapter 6), and the user

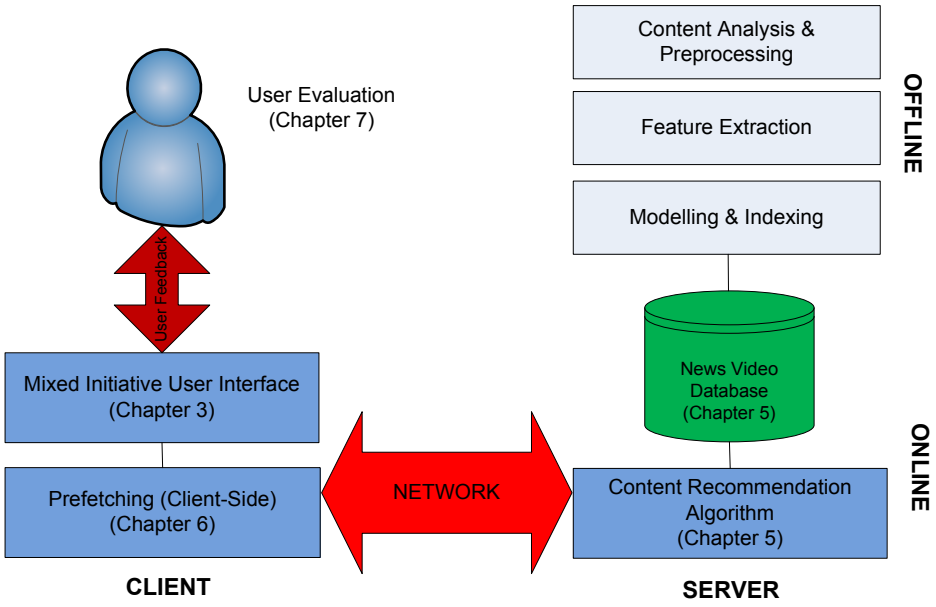


Figure 1: Multimedia information retrieval system structure & research scope

experiments and behaviour studies (Chapter 7). The entire system architecture is addressed in Chapter 4.

1.2 Organisation

This dissertation is thus organised as follows:

Background A brief review of the basics of multimedia indexing, relevance feedback, video streaming and methods for network delivery of multimedia are covered in this chapter.

Mixed-Initiative UI for News Videos In this chapter, the new approach involving the interface designs and usage scenarios are described. The interface design considerations and news usage patterns are discussed.

System Architecture & Model The generalised client-server system architecture and system model used for video delivery to a mobile device is first defined, that will be used subsequently for simulations and user subjective experiments. In particular, the feedback loop and different time durations during the retrieval process are detailed.

Content Recommendation Algorithm The video database content and indexing method are first described, before delving into the relevance feedback style algorithm, mathematics, and corresponding simulation results and discussion.

Prefetching for Mixed-Initiative Video This chapter further details incorporating prefetching of the video prefixes in the system and discusses various prefetch strategies, tradeoffs and the subsequent results.

User Evaluation The user experiment protocol and overall results are presented, alongside a discussion of the user issues and performance of the real mobile system.

Conclusion Finally, the dissertation concludes with a summary of the research contributions and future directions.

CHAPTER 2

BACKGROUND

This chapter provides a brief overview of the fundamentals of multimedia retrieval, RF, and general system architectures for multimedia delivery, necessary to construct a MIR system for mobile devices.

2.1 Multimedia Information Retrieval

The value of information is determined by how easily it can be accessed and retrieved. Recent years have seen burgeoning digital media archives and therefore immense research interest in developing techniques to facilitate searching and retrieval of desired information from this large collection of multimedia data. However, understanding and searching the diverse and rich content of a multimedia collection still remains a difficult task.

Multimedia information retrieval (MIR) is the science of searching for multimedia content and the information contained within them, making such content easily accessible. It is interdisciplinary, drawing from the areas of computer vision, machine learning, databases, cognitive psychology, linguistics, statistics, and others. Famous commercial examples of IR systems are the search engines Google [18], Microsoft's Bing [19] , and more historically, AltaVista.

To easily access and retrieve multimedia data, the content must first undergo an offline pre-processing phase, before being placed in the online phase for search and access, as depicted in Figure 2. During the offline phase, semantic information or knowledge of the multimedia content are first human annotated and/or extracted using automated methods into a suitable format and representation. The extracted representations of the multimedia data's content are known as features, such as keywords and text, colour, shape, motion vectors or image difference. Sometimes, models

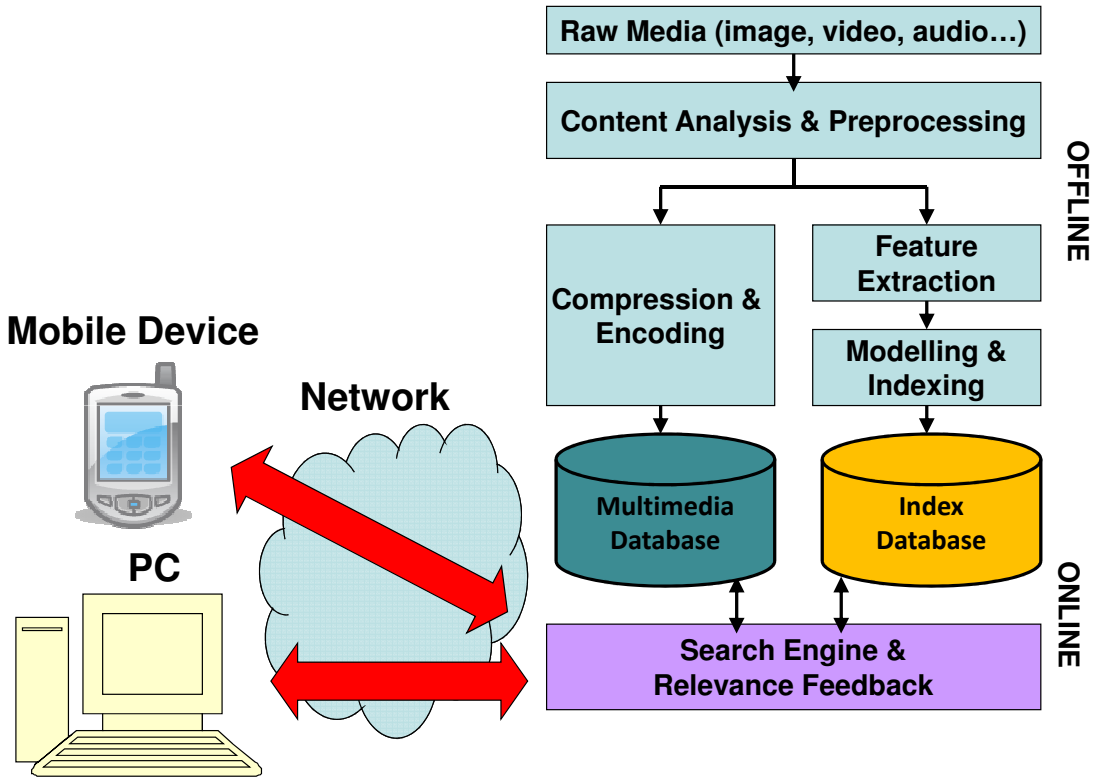


Figure 2: Typical multimedia information retrieval system structure

may be developed and used to further provide domain-specific structure or meaning to the representation.

The multimedia data is then indexed using the given representation and stored in a database. For efficient access to items in the database, indexing methods such as inverted indices, B-trees, and R-trees are sometimes used. During the online phase, users submit a query to the system, and multimedia data that has high semantic relevance (similarity) that matches the query is retrieved from the database and delivered to the user. The user then refines the query interactively on the basis of the retrieval result.

Currently, MIR systems attempt to move away from textual search engine style query and ranked result displays to more interactive search styles and agent interfaces. RF attempts to interpret continuous repeated feedback from the user toward learning more about the user's query [20, 21]. Another perspective is that semantics is an emergent property of the interaction between the user and the system and is grounded in context and actions of the user [22]. Therefore the meaning of the data will be revealed by interpreting the sequence of queries posed by the user.

2.1.1 Feature Extraction & Representation

In order to build representations that accurately capture the meaning of the multimedia content, features must be identified and extracted from the content. Depending on whether the content is a text document, image, video or audio, features could be keywords, text, term (word) frequency, colour, shape, motion vectors, image difference or other descriptive values. These can be extracted automatically using optical character recognition (OCR), computer vision or automatic speech recognition (ASR) techniques or annotated manually by a human. To minimise computational cost and reduce the dimensionality of data, the smallest and richest feature representation is desired.

Most models represent a multimedia data item or object as a n-dimensional vector

of numerical features X, where:

$$X = (x_1, x_2, \ldots, x_n)$$

In the vector space model of information retrieval [23], documents and queries are represented by term feature vectors, where each feature in the vector represents the frequency of occurrence of a word. It is sometimes used together with tf-idf (tf-idf) weighting to determine the relative importance of the words in the document. Cosine similarity is commonly used as a measure of similarity between two vectors of n-dimensions by finding the cosine of the angle between them. The most similar documents are then returned.

2.2 Relevance Feedback

Relevance feedback (RF) is a technique that supports user interaction in IR, attempting to integrate continuous repeated feedback from the user toward learning more about the user's query [20, 21]. During the RF process, the retrieval system learns or estimates the user's interests based on user's judgment of relevance or non-relevance of the displayed items in order to refine the query. Figure 3 illustrates a typical information retrieval and RF system UI for image or video retrieval on a desktop. The user is usually shown a list of candidate items, and is asked to decide whether each item is relevant or irrelevant. There are two different data sets: relevant items and irrelevant items. The parameter, semantic, or search space is then modified to reflect the given relevant and irrelevant examples, and a new set of candidate items is then displayed to the user. This process can be repeated several times in order to produce the best results.

Relevance feedback (RF) was originally developed for textual information retrieval [24, 25, 23]. In the vector space model of textual IR, documents and queries are represented by term feature vectors. The concept of RF in the vector space is to

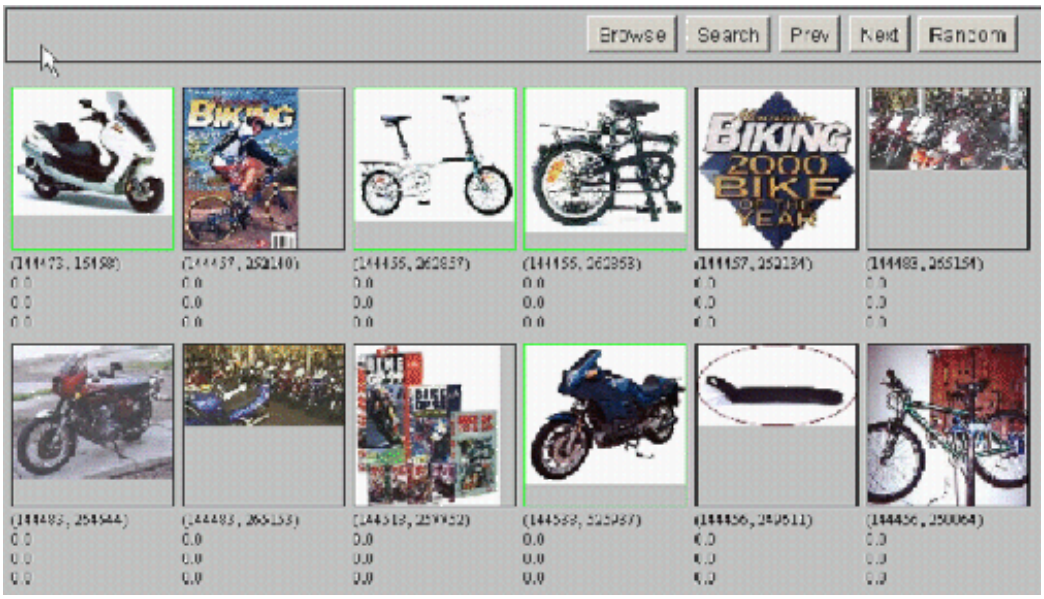


Figure 3: Typical image-based relevance feedback system interface [1]

estimate the ideal query point by moving the current query vector toward positive feedback and away from negative feedback based on the cosine similarity [24] as depicted in Figure 4. The next modified query point Q_m is then updated based on Rocchio's formula [25]:

$$Q_m = aQ_0 + b\frac{1}{|Dr|}\sum_{D_j \in D_r} D_j - c\frac{1}{|D_{nr}|}\sum_{D_k \in D_{nr}} D_k$$

where Q_0 is the original query vector, D_r and D_{nr} are the set of known relevant and non relevant documents respectively, and a, b, and c are weights attached to each term.

Therefore, the basic framework of RF can be based on machine learning or parameter estimation. These RF algorithms tend to have a simpler model for semantic similarity to the user's query. There is an assumption that all relevant items are clustered together, and they may not uncover the other concepts that may emerge during the user's interaction. A violation of this property could be several clusters of relevant items that do not have the same feature overlap.

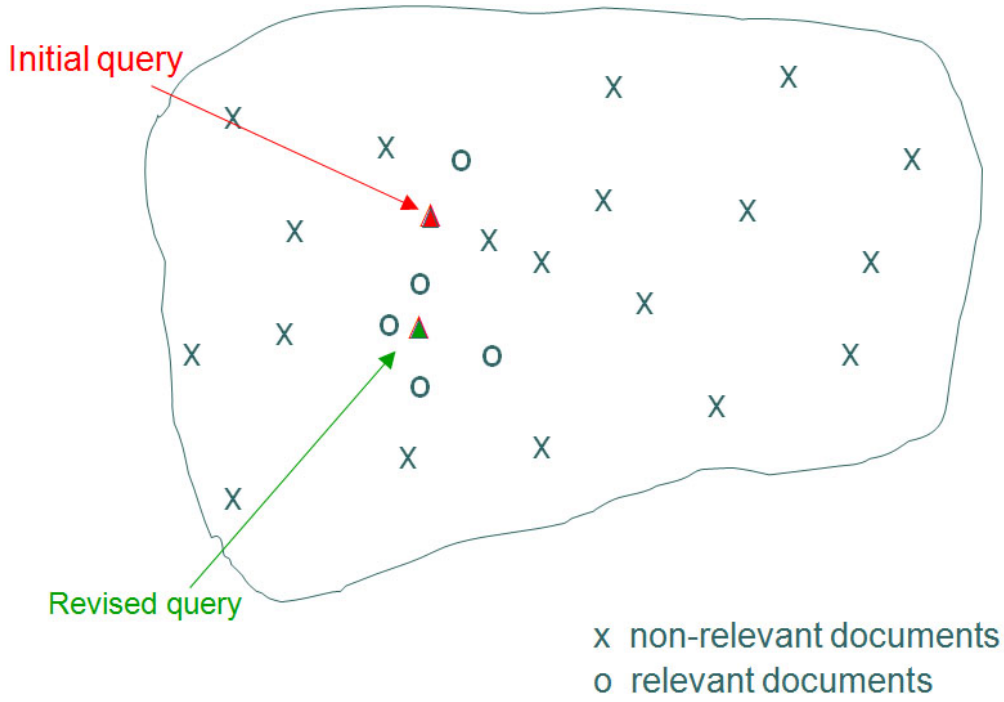


Figure 4: Query refinement in a vector space relevance feedback system [2]

Another way of viewing RF is to model it as a particular type of pattern classification in which the positive and negative examples are found from the relevant and irrelevant labels respectively. Thus, it may be possible to apply any learning algorithm into the RF cycle.

There are a few important considerations: The training data set may be too small as it is constructed by user feedback. Also, since a user interacts with the system in real-time, the learning process must be performed fast, and a complex learning process is not desirable. Various issues and considerations in the design of the RF mechanism are further discussed in [26].

2.3 Implicit Feedback

Relevance feedback (RF) methods typically require that users explicitly give feedback by specifying keywords, selecting and marking relevant and irrelevant items, or answering questions about their interests, amongst other methods. Such relevance

feedback methods force users to engage in additional activity. As the cost to the user is high, the user is sometimes reluctant to provide this additional feedback.

An example is users abusing the rating scale on the popular video viewing website YouTube [27]. Users are allowed to provide star ratings indicating how much they like a video. Five stars for a great video, and one star for a hated video. It was discovered that the user given ratings were mostly all or nothing. Great videos encouraged action, and anything less prompted indifference, thus defeating the purpose of the scale itself [3]. It can therefore be difficult to collect the necessary data from users, and the feedback method should be as simple as possible.

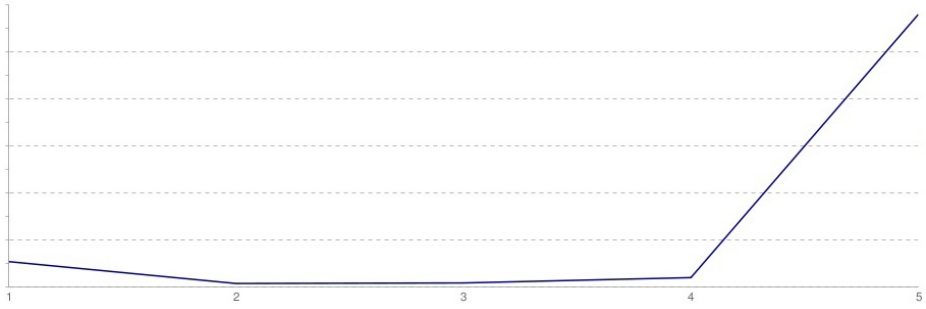


Figure 5: YouTube 5 star ratings distribution [3]

Literature has explored implicit feedback techniques for user profiling and query expansion in the context of web browsing and information retrieval tasks [28, 29, 30]. These techniques unobtrusively obtain information about users by watching their natural interactions with the system. Some of the user behaviours that have been most extensively investigated as sources of implicit feedback (in the context of web browsing) include reading time, saving, printing and selecting [31].

The primary advantage to using implicit techniques is that they remove the cost to the user of providing feedback. In addition, implicit measures can be combined with explicit ratings to obtain a more accurate representation of user interests. Implicit feedback techniques have been used to retrieve, filter, and recommend a variety of items such as hyperlinks, web pages and email [32]. It may be beneficial to define

and select a certain subset of behaviours in the mobile context and apply them to the recommendation algorithms to enhance the user experience and improve recommendations.

2.4 Collaborative Filtering

Recommender systems improve access to relevant data and information by making personalized suggestions based on previous examples of a users likes or dislikes. Most existing recommender systems use collaborative filtering techniques [33, 34] that base the output recommendation on other users likes and dislikes - a form of computerised matchmaking. An example of this is the recommendation service of the online bookstore Amazon.com [35]. Recommendation services are also used in many other domains such as movies (Netflix [36]), and music (Pandora [37] and Last.fm [38]).

The system maintains a database of the preferences of individual users, and finds other users whose known preferences correlate significantly with a given user and recommends to this person other items that were liked by his/her matched users. Several schemes to compute similarity between users have been proposed, such as the Pearson correlation coefficient [39]:

$$w(a,i) = \frac{\sum_j (v_{a,j} - \bar{v_a})(v_{i,j} - \bar{v_i})}{\sqrt{\sum_j (v_{a,j} - \bar{v_a})^2 \sum_j (v_{i,j} - \bar{v_i})^2}}$$

and Cosine distance (adapted from information retrieval):

$$w(a,i) = \sum_j \frac{v_{a,j}}{\sqrt{\sum_{k \in I_a} v_{a,k}^2}} \frac{v_{i,j}}{\sqrt{\sum_{k \in I_i} v_{i,k}^2}}$$

Collaborative methods assume that a given users preferences are generally the same as another user of the system, and that a sufficient number of votes have been collected. There needs to be enough other users already in the system to find a match, if not it tends to suffer from a cold-start problem [40]. Collaborative approaches also

have a popularity bias; they tend to recommend popular items, and fail to recommend items that nobody has yet seen or liked. Compared to information retrieval methods, collaborative filtering is not content-based and does not use information about the item itself to make recommendation suggestions.

More recent work has attempted to move toward content-based retrieval methods for recommendations [41] and to integrate collaborative information. Results point to the feasibility of such a hybrid method that overcomes the various disadvantages discussed above.

2.5 Video Delivery Over Networks

The main approaches of *downloading* and *streaming* video over packet networks such as the Internet and their associated challenges are discussed in this section.

The first most straightforward method for video delivery is *downloading*. The entire video file is treated like a regular file download, and allows the use of established protocols such as Transmission Control Protocol (TCP) at the transport layer, or Hypertext Transfer Protocol (HTTP) or File Transfer Protocol (FTP) at the application layer to handle the transfer. However, compared to the average file size, the video file size is usually extremely large, in the region of megabytes or even gigabytes.

The obvious practical disadvantage is this method requires long download times and a large storage space. The other issue is that the entire video must be downloaded before playback can commence. This tests the user's patience, and becomes inconvenient when evaluating a video's content. If the user is unsure of whether to watch a video clip, the video must first be entirely downloaded before being viewed and only then can a decision be made.

Streaming is another method of video delivery, and is intended to overcome the existing problems with downloading and also provides additional flexibility in viewing. The fundamental concept of streaming is to separate the video file into parts and

transmit the parts one after the other. The receiver is able to decode and play back the video as these parts are received without waiting for the entire video file [42]. There is usually a short pre-roll delay of 5-15 seconds between the start of video delivery and the start of playback. This is done to fill the *playout buffer* to overcome delay jitter, details of which are discussed subsequently in this section.

The main benefit of streaming a video is that simultaneous delivery and playback of the video is possible. There is only a short waiting period before viewing can start, and there are low disk storage requirements since only a small part of the entire video is stored at the client-end at any point in time. The time duration of the playout buffer determines the length of the delay, and the amount of the data in the buffer determines the required storage size.

2.5.1 Challenges in Video Streaming

Issues that affect video streaming include varying and dynamic values of *bandwidth*, *delay jitter* and *loss rate*. These problems exist since only a best effort service is provided over the Internet.

Bandwidth is defined as the maximum amount of information (bits per second) that can be transmitted along a channel, i.e. the effective data transmission rate. The available bandwidth between two points is generally variable and sometimes unknown. If the sender transmits faster than the available bandwidth, congestion occurs and packets are lost. The result is a poor viewing experience at the receiving end, with pauses mid-stream, jerks, and corrupted or lost frames. Streaming is not possible at a higher bandwidth than what is available. If the sender transmits at a slower rate than the available bandwidth, the receiver sees less than optimal video quality than what is possible. The idea is to estimate the available bandwidth and subsequently match the transmitted video bitrate to the available bandwidth as far as possible.

The time taken for a packet to travel end-to-end may fluctuate greatly. This variation in end-to-end delay between packets is known as *delay jitter*, or more precisely,

packet delay variation (PDV). This causes problems as the video frames must be received, decoded and displayed at a constant rate. Late arriving frames due to delay jitter results in poor video playback, such as frequently stalling and jerky video. Delay jitter can be mitigated by selecting a properly sized playout buffer at the receiver end. While this playout buffer compensates for jitter, it introduces an additional humanly detectable startup delay at the beginning of video playback and thus affects the user experience. Usually, 5 to 15 seconds of the beginning of a video is buffered before playback commences. Buffering also requires some additional storage space at the client end.

Despite the introduction of a user perceptible delay before playback, buffering brings about certain advantages due to the extended presentation deadlines for the video frames. As described above, delay jitter is reduced or possibly eliminated. There is also error recovery through retransmissions, when packets are lost, and smoothing of throughput fluctuation. During playback, the buffer is depleted first, and allows time for lost packets to be retransmitted, and sustains streaming during times when throughput is low. Packet losses can occur due to congestion, bit errors or burst errors.

There are additional challenges when considering both wired and wireless links in the streaming path. There is a much longer packet delivery time with the addition of wireless links, with the round-trip propagation delay in a 3G wireless network in the order of 100 ms. There is also difficulty in determining network conditions using end-to-end measurements, as there are frequent packet losses and packet corruption. Packet losses, loss rate, rate control and other streaming issues are further discussed in [42].

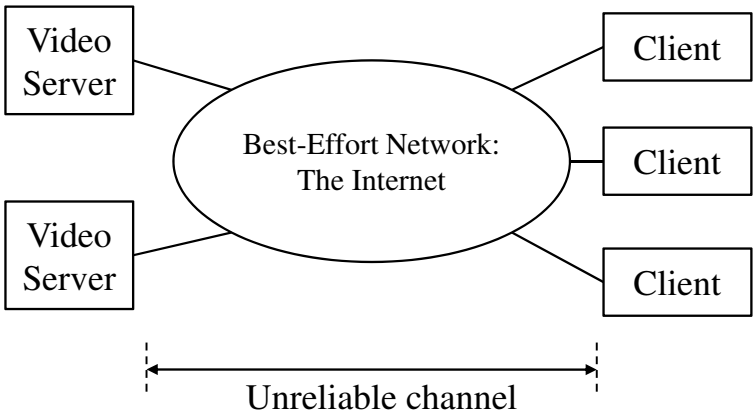


Figure 6: Typical client-server network

2.6 System Architectures for Network-Based Multimedia Retrieval

The most prevalent architecture for multimedia access over best effort networks is the *client-server paradigm* as illustrated in Figure 6. Client-server computing is a distributed application architecture in which the workload is separated between the *server*, which provides a service, and the *client*, which requests a service. Clients and servers typically operate on separate hardware interconnected by a computer network, and server hardware is usually a high performance host that can share resources with many clients. Interaction between clients and servers proceed according to a protocol.

Many services on the Internet are client-server applications such as web hosting (HTTP), file transfer (FTP), DNS and finger. In the case for MIR, the multimedia is stored on the server, and the server listens for a client request. When a request is received, the algorithm is executed at the server to find the most matching multimedia items. The matching items are then retrieved and delivered to the client.

In contrast, in peer-to-peer architectures, each host can simultaneously act as both a client and a server, and each has equivalent responsibilities and status, without

a central coordinating host. This paradigm was popularised by illegal file sharing networks such as Napster and BitTorrent for the distribution of large media files.

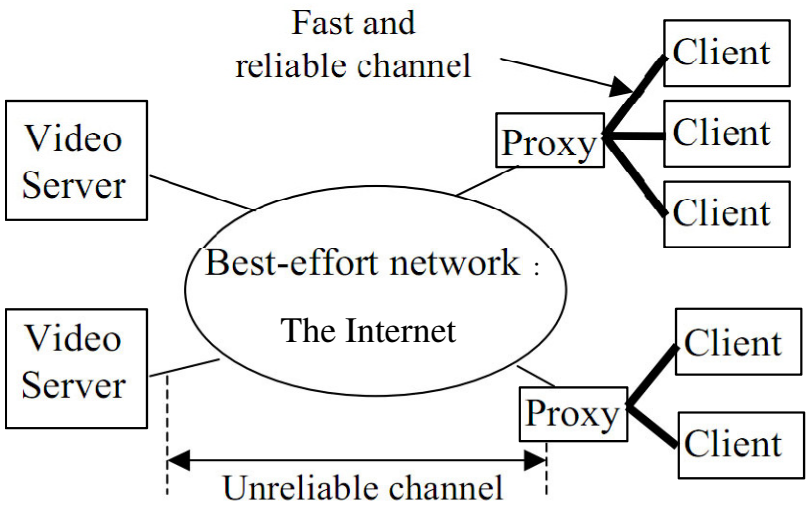


Figure 7: Proxy caching

To accelerate web browsing and reduce networking costs, proxy caching is sometimes deployed on the local network. A caching proxy server accelerates service requests by retrieving content saved from a previous request made by the same client or even other clients. Local copies of frequently requested content are kept in anticipation of requests, reducing bandwidth usage and cost while significantly increasing performance, especially when large multimedia objects are accessed frequently. This is illustrated in Figure 7.

CHAPTER 3

MIXED-INITIATIVE UI FOR NEWS VIDEOS

This chapter defines the interface design problems for mobile devices, and explains the mixed-initiative approach intended as the basis of the UI-centric solution for the system.

3.1 User Interface Considerations for Mobile Devices

Discussion on suitable user interface designs for mobile devices has persisted in recent literature, with the general consensus that a mobile interface should require a different interaction style from that of a "window, icon, menu, pointing device" (WIMP) style desktop interface. Attempting to cram all the functionality of desktop system into a mobile device is not possible, as small display sizes, awkward methods of data input and distractive environments have been identified as major constraints.

Interaction-wise, the abovementioned mobile device constraints lead to users having vastly different viewing styles. Mobility means activity spurts interleaved with quiet, unstructured periods of time also due to the strong influence of distractive environments such as being on the subway or waiting for the bus. Overall, viewing was low commitment and transient, with short attention spans, and with extremely short interaction intervals [5]. Average viewing time is around ten minutes long.

Some solutions are to minimize user input, predict and filter information, reduce memory load and use different modalities and interaction methods. To overcome these limitations, recommending and filtering methods can be used to deliver the relevant videos to the user. Although there currently is no established methodology on which to base an interface design for a mobile device, existing literature [5, 7, 8] has proposed some rough design guidelines:

- Determine what kind of information a user wants to access via the mobile device

and what style of interaction a user will be using, in order to have most benefit

- Minimise user input - where applicable, provide simple user selections such as yes/no options, instead of asking the user to articulate input queries or use visually demanding browsing that requires careful inspection of the screen

- Filter out information so that only a small amount of the most important information can be quickly and readily accessed via the mobile device rather than trying to provide full coverage of all information

- Use a layout that does not require a large space, e.g,. converting spatial information into a temporal format

The guidelines point towards more pre-processing on the system and server side to determine which pieces of information a particular user will most likely to want to see. Thus, it naturally leads to the development of systems which proactively recommend particular pieces of information to the user, and consequently demand less intensive interaction on the user's part.

Despite the constraints of small screens and limited input methods, mobile devices have vastly different affordances, and often involve location and context. A mobile device is also a highly personal communications device and ownership is usually restricted to an individual. The latest devices have cameras, touch screens, accelerometers, GPS, Bluetooth and light detectors. This creates an emerging mobile computing environment where tactile and visual input is accepted, and information about an individual user's location and physical environment are provided.

Mobility also brings about new social norms and emergent behaviours, such as continuous change in the user context and spontaneous and impromptu information requirements of the user. Thus, the challenge is to create more sophisticated direct manipulation interaction methods and cater to each user's constantly changing situation and query requirements.

3.2 News Consumption Patterns

In terms of print and webpage full length news documents, most readers first scan the news for headlines and current breaking stories, before focusing on certain articles for depth. The news supply is similar for both online and print, and studies have found few differences in consumption between online and print versions of news. Attention to news stories varies depending on news category, reader gender and interest in a particular topic rather than whether the format appears in print or online [43]. There is also little evidence of a consistent news reading pattern whether online or print, and print version readers do not read more than online news users.

Additional studies have also focused on news consumption via new media channels. In the Associated Press' multinational anthropological study on how the younger population consume news, there appears to be a strong appetite for news among younger audiences. The younger set use news as "social currency", and consume it in a different way than mainstream media is prepared to deliver it. They check news multiple times a day and use a variety of methods such as e-mail, web and mobile devices apart from the mainstream media channels of newspapers, television and radio. New methods of news delivery such as email and mobile were favoured over mainstream media channels.

The observation is that with these faster delivery vehicles and platforms, a news model based on quick delivery and quick-scan consumption is created. Bite-size pieces of news in the form of headlines and the latest breaking news are quickly consumed. However, this new generation of news consumers seldom dig deeper or "below the fold". This group claims they are soon inundated with facts and updates and showed signs of news fatigue from information overload, especially when the information stream mostly present recycled headlines and updates. Despite this, more information is craved in the form of a news story's depth. These consumers desired back stories, future stories and spinoffs, aspects of a story they were unable to find easily [44].

3.3 Traditional TV Watching

The interaction style of traditional TV watching differs from mobile video viewing in several aspects. In general, television watching is generally a passive, "lean back" activity that takes place at home, whereas a mobile video viewing may occur almost anywhere, especially during impromptu and unstructured down time. TV may be watched in a group socially with friends or family, however such viewing is primarily a serial and solitary activity on a mobile device. TVs have also have remote controls that facilitate channel switching in search of suitable content.

Viewing commitment also varies with traditional TV. There are higher levels of commitment with favourite TV programmes or pay-per-view where users prefer not to be interrupted. Lower commitment viewing exhibits more channel switching and competes with other activities in the home, or the television might just be playing in the background [5].

Compared to a networked mobile device, traditional TV is defined in [5] to have certain features:

- **Instant on**: Once the TV is switched on, the content is received immediately.
- **Continuous**: Content is shown all the time. Once a program ends, another begins.
- **Seamless and easy switch**: Switching channels is instantaneous.
- **Graceful Transitions**: Transitions from one program to the next is smooth as the content is edited and selected by broadcasters.
- **Reliable service**: TV broadcast interruption or halting is very rare. A television breaking down is also uncommon.

These features of traditional TV contrast with current technology for streaming video onto mobile devices. As detailed in section Section 2.5, delays due to video

buffering or packet loss affects the user's perception of mobile videos as an instant and reliable service. Mobile video does not start instantly, and there is a buffering delay when switching between videos. Therefore steps should be taken to ensure that mobile video watching reliability and interactivity is similar to, or on par with traditional TV.

3.4 Multi-Modal Mixed-Initiative User Interface Design

Mixed-initiative[1] describes a method whereby intelligent services and users may collaborate efficiently to achieve the user's desired goals [12, 13]. In such a model, there is a natural dialogue and interleaving of contributions from both the user and automated service that converges on the problem. It can adapt sessions to the user, deal with interruptions and help manage the user's focus of attention. These new interface models provide a starting point from which to develop compelling multimedia retrieval applications.

For mixed-initiative interaction to be effective, the system must provide significant value-added automation over solutions with just direct manipulation. It should consider uncertainty about the user's intentions, and make decisions about when to engage users, how to best contribute and when to pass control back to the user. Preferably, a working memory of the recent interactions should be maintained and the system learns though observation of the user. More importantly, poor guesses should be minimized and direct invocation and termination should be allowed. This suggests that machine learning and probabilistic user models are useful to reason about the user's intentions and requirements.

Based on the principles of mixed initiative interfaces, an adaptive TV-channel-like metaphor is proposed to display recommended news video clips temporally. It can be

[1]Although it has been often used in the context of interface smart agents, it can refer generally to methods that support a synergistic interleaving of contributions from the user and automated services with the goal of arriving at solutions

thought of as a single multi-modal news channel that adapts to the user's likes and dislikes, if the user decides to intervene.

The overall UI is designed to be as simple as possible and assign the the majority of the screen estate for video display. For non-touchscreen-enabled devices, the number of elements shown on-screen are restricted to the three buttons corresponding to the binary voting operations of *vote-up* and *vote-down*, with an additional *skip* button at the edge of the screen as seen in Figure 8. The device can be oriented vertically or horizontally. The system can consist of three primary modes, *manual*, *automatic* and *semi-automatic*. When the client application is launched on the mobile device, one of the latest video clips is selected and immediately starts playing. During video playback, the user has a choice to either continue watching the video, or press one of the buttons corresponding to one of the operations. Each video clip will only be shown once and will not be repeated.

3.4.1 Manual Mode (Pull)

The user is able to invoke the manual video selection mode from the menu at any time, showing a list of current video clips in reverse chronological order. The action of manually selecting a video clip from the list to watch is interpreted as positive user feedback to the system. The user feedback is used to make recommendations when in the semi- or fully automated modes. This can be seen in Figure 9.

3.4.2 Automatic Mode (Push)

Should the user decide not to intervene via voting, the system switches to the automatic mode, advancing from one video clip to the next, much like a regular TV news channel. If the recent voting feedback is not available, the system selects a video at random from the current list of latest news videos. This behaviour is similar to passively watching a news channel on a traditional TV.

Figure 8: Implementation on the HTC Magic (MyTouch)

3.4.3 Semi-Automatic Mode (Push-Pull)

This is the most important component of the synergistic user-machine feedback system. The user is presented with one video clip at a time, and the operations are limited to the simple binary voting system: *vote-up*, *vote-down* or a separate choice to *skip*. The vote-up and vote-down operations translate as explicit positive or negative feedback to the system, and each video clip can be voted on only once.

After a vote-up, the currently playing video clip continues playing, since it is assumed that a user would prefer to continue watching. The user then presses skip to move to the next video. Initiating a vote-down immediately terminates the currently playing video and advances to the next video clip after registering the feedback.

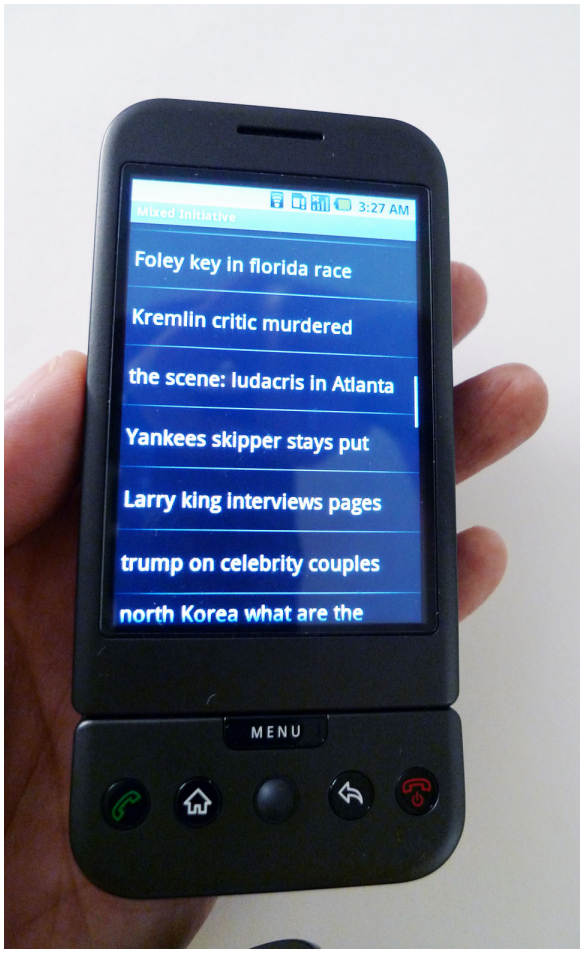


Figure 9: Manual video selection mode on the HTC Dream (G1)

Skip advances to the next video without providing explicit feedback. The voting functions as a type of relevance feedback mechanism, to determine the next clip to be recommended. There is no explicit assumption of users self reporting their interests before starting to interact with the system.

Example: (a) The user is first presented a clip on the Mark Foley scandal. The vote-down operation is performed, before proceeding to the next clip. (b) The next clip is a story on the North Korean nuclear test, which is watched till the end and voted up. (c) Subsequently, another clip on the Foley scandal is presented. The user performs the vote-down operation again. The system takes note of user's negative preference for Mark Foley news, and subsequent videos (d) and (e) are not shown

(a) (b) (c) (d) (e)

Figure 10: Semi-automatic user voting example

having such content. Figure 10 illustrates this process.

As there is a possibility decreasing or changing interest with the amount of similar news which have been watched, the filtering fades after a threshold of similar videos are watched and other recent videos are then shown. Each video will only be viewed once, and there are no repeats unless manually selected. However, the system can be reset by selecting an option in the menu.

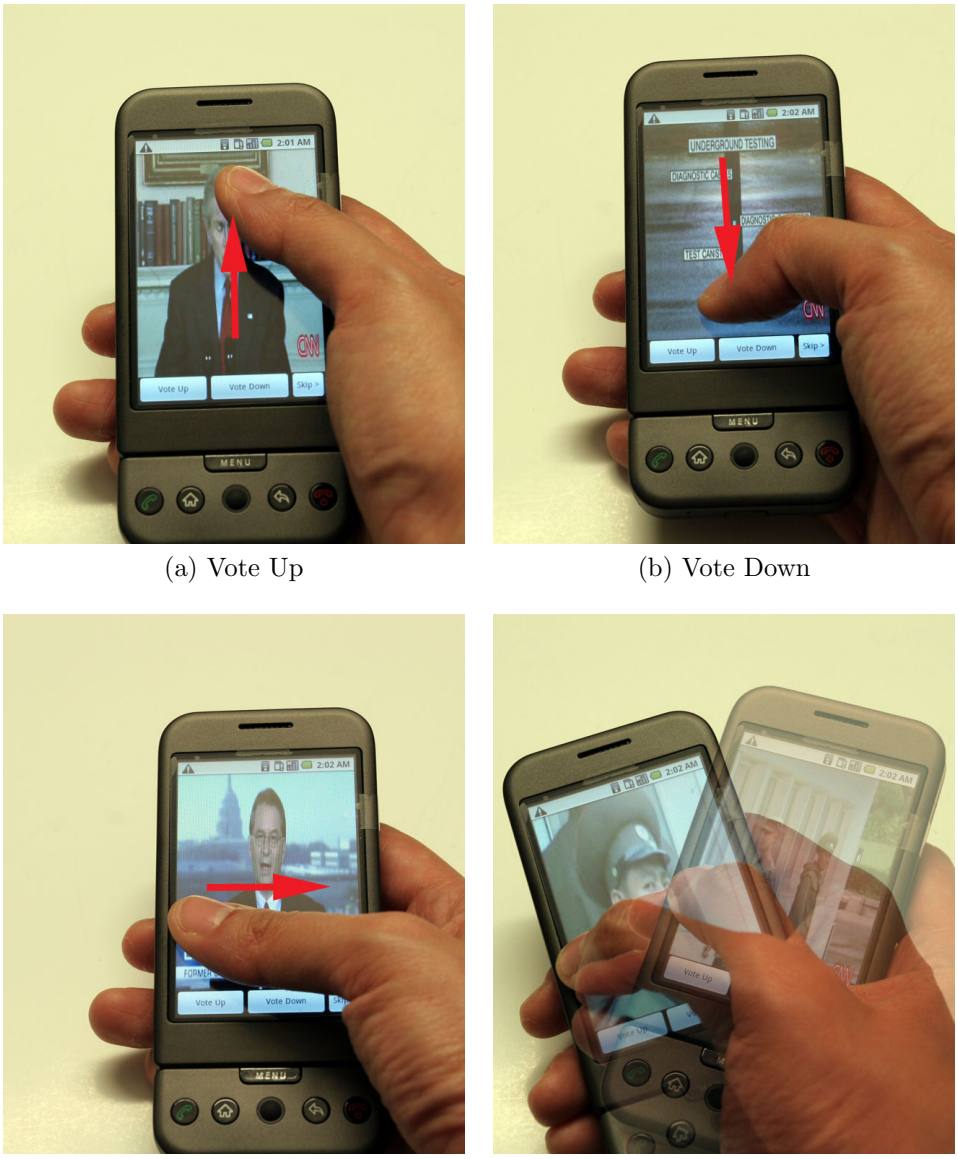
3.4.4 Alerts & Dialogs

If newer video clips become available in the database, these will be presented subsequently in sequence to the user in the semi-automatic or automatic modes, or appear in the list in the manual modality. For particular headline and breaking news that become available, a dialog is displayed as soon as viewing of a clip has concluded, and indicates the importance of this particular piece of news. The user has a choice of whether to continue watching this video, or dismiss it.

3.5 On-Screen Gestures

Each of the operations has a additional corresponding on-screen finger gesture – *vote-up* is a finger flick upwards, *vote-down* is a finger flick downwards, *skip* is a finger flick rightwards (or leftwards) and shaking the device resets the votes and video sequence (Figure 11). With on-screen gestures, it is possible to use the entire screen for the video display, and the operations are easily executed with one hand holding the device

and the thumb on the screen.



(a) Vote Up

(b) Vote Down

(c) Skip

(d) Reset

Figure 11: HTC Dream (T-Mobile G1) with on-screen gestures

CHAPTER 4

SYSTEM ARCHITECTURE & MODEL

This chapter presents the system model, concepts and assumptions behind the design. This is then converted to the system architecture and the constituent components. The processing of queries and possible mobile parameters are also described.

4.1 System Model for Mixed-Initiative

As an IR system is large and consists of many interacting components, a model is first developed for simplicity and to bring focus to the main parameters pertaining to video content retrieval to the mobile device and user relevance feedback. This allows the development of simulated user scenarios and a study of the system's behaviour.

A client-server model is assumed, as this is the most typical model for video distribution. There are many delivery models and content associated with mobile video watching, from "live" streaming to pre-recorded. In this case, it is assumed that the video will be streamed to the clients, not downloaded. However, the video is not "live" streaming when compared to an actual TV news channel, but rather, pre-recorded and edited individual news video clips approximately five minutes long. These video clips are assumed to be continually updated on the server periodically throughout the day and sorted by recency. The news video clip database is described further in Section 5.1.

In such client-server interactive sessions, a large amount of multimedia data has to be delivered through the network from server to client. In video streaming, there is a startup delay to buffer video before playback. Therefore, the user experience may be unsatisfactory despite high semantic quality when relevant videos are being delivered. Especially for mobile devices, a lower bandwidth wireless link will be a bottleneck to interactivity, and the user perceived latency will be higher. The available bandwidth,

video bitrate and variable delays in a best effort packet-based network affect this user perceived latency. Exploring how to improve both the relevancy and user perceived latency are further discussed in the later chapters.

While the video is being streamed, the user response for RF needs to be transmitted in the opposite direction, and this response data size is relatively small compared to the video data. Server processing time and round trip time (RTT) are generally negligible. This system model is illustrated in Figure 12.

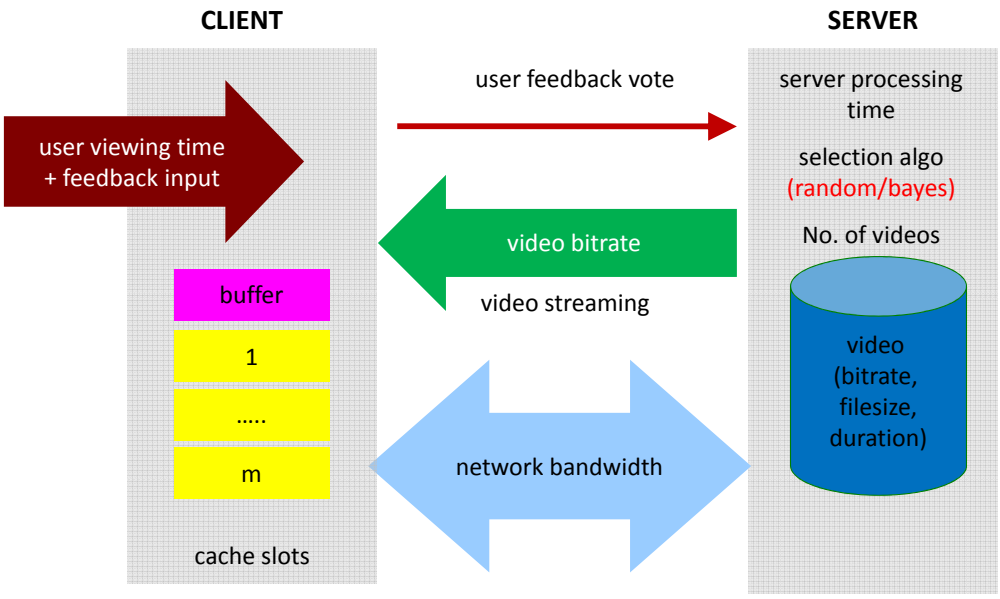


Figure 12: System model

4.2 Timing Definitions

Assuming the streaming method of video delivery, Figure 13 illustrates the various time durations during the access and viewing of one video clip by a user of the system during an iteration in a video viewing session. Each video viewing session can consist of multiple iterations of getting a video, and submitting a user vote as feedback. This

feedback vote can be an up or down vote, and the timeline is illustrated in (a) for an up vote and (b) for a down vote.

The *server processing time* is the time taken to decode and interpret the message from the client, compute the next best video to recommend, encode the reply packet and transmit this packet to the client. The *user feedback processing time* is the time taken to collect the user's voting feedback, encode the packet into a message for the server, and transmit this packet to the server. Both the server processing time and user processing time are considered negligible in terms of duration compared to the larger time to buffer and then watch the video.

Streaming a video introduces a perceptible delay before video playback, where 5 to 15 seconds of the beginning of a video is buffered before playback begins. This is defined as the *buffering time*. In an entire viewing session, there will be pauses in between video clips due to buffering.

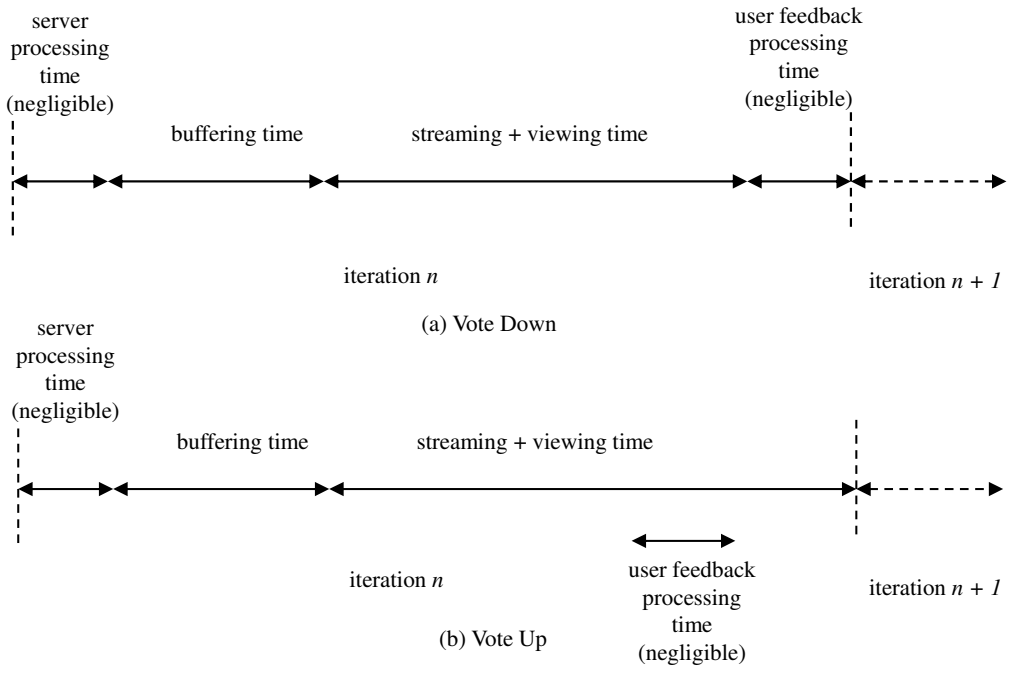


Figure 13: Timing diagrams

4.3 Mobile Device Specifications

Most mobile devices can be roughly classified into two groups, the lower end basic phones and PDA devices, and the higher end "smart" phone that have sophisticated operating systems and support for multiple applications and programming models. Examples of "smart" high end phones are the iPhone, Blackberry and Android devices. The specifications are summarised in Table 1 below. These will be used as a basis for the system design and experiments in the subsequent chapters.

Table 1: Typical mobile device specifications

Specification	**"Low-end" phone**	**"High-end" phone**
Screen resolution	128x96, 176x144	320x240, 320 x 480, 800x480
RAM	8 - 32MB	128 - 512MB
Storage (disk)	16 - 128MB	256MB - 16GB
Video bitrate	5-40Kbps	50 - 500Kbps
Video framerate	4-9 fps	15-30 fps
Video codecs	MPEG4, H.263, H.264	
Bandwidth (theoretical)	56 Kbps (GPRS) 144 Kbps (CDMA2000) 237 Kbps (EDGE) 384 Kbps (UMTS) 14,000 Kbps (HSDPA)	

Currently, typical mobile devices currently have a 320x240 screen in 16-bit colour, 128MB RAM and 256MB to 4GB storage. In practice, the actual bitrates of mobile networks are usually around half of the theoretical maximum specified, therefore lower network speeds of between 56 to 384 Kbps are usually the norm.

The Google Android phones used for the implementation and experiments have the following specifications: 320x480 resolution, 192MB RAM, 512MB onboard storage, up to 16GB expandable storage and support for Wi-Fi, GPRS, EDGE and UMTS. These specifications correspond to a higher end phone in today's definitions.

4.4 System Architecture

Given the system model and device specifications, it can then be translated to the system architecture and the constituent components. This system architecture is described in Figure 14.

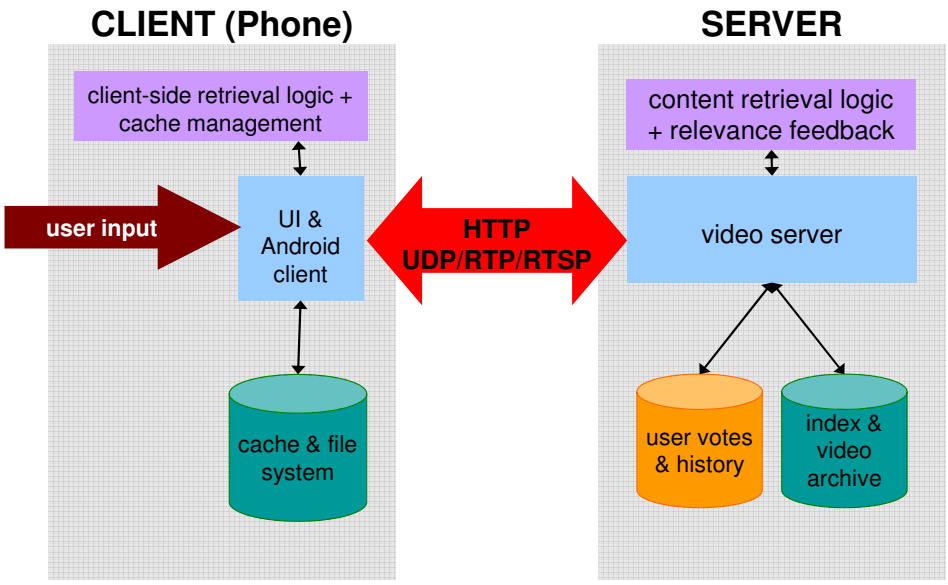


Figure 14: System architecture

On the server-side, the videos are stored on disk and indexed for easy access. Any collected user history or votes are also stored. This will probably require an indexing method that is custom to both the user votes and video archive. A typical server configuration running a Linux operating system such as RedHat with Apache for the web service or Windows and IIS can be used. The content retrieval and RF logic can be implemented using any server-side language such as Perl, Python or Java, supported by the specific web server used.

At the client-end, the UI and video displaying components are implemented, along

with any client-end retrieval logic and prefetching cache management logic. Examples of possible client platforms are iPhone OS, Google Android or Symbian. Any prefetched or buffered items are stored on disk or in memory on the mobile client.

Communication between server and client can be handled using JavaScript Object Notation (JSON) over HTTP. JSON is a lightweight data-interchange format, consisting of data structures with name value pairs and are supported in most languages for client-server communication. This can be used to relay messages such as the client's request to the server for the next video, or the user's feedback response to the server.

The video streaming itself can be handled using a transport layer protocol, such as User Datagram Protocol (UDP), as the requirement is time-sensitive and there is no absolute need for reliability guarantees. Losing packets is preferable to waiting for delayed packets. Real-time Transport Protocol (RTP) together with Real Time Streaming Protocol (RTSP) can also be used as a specific protocol to handle media streaming to provide quality of service and media specific jitter compensation and out of order detection.

4.5 Implementation

The initial client was implemented as a PC-based application using the Java Developer Kit (JDK) with the Java Media Framework (JMF) on Windows operating systems. It was designed to prototype the necessary functionality and deliberately displayed the video in a small window on the desktop. The user voting operations are displayed as icons on a toolbar located near the top of the window as seen in Figure 15.

The current prototype client interface and application is implemented in Java on the HTC Dream (T-Mobile G1) and HTC Magic (MyTouch) running the Google Android mobile device platform (with the 1.6 SDK "Donut") as seen in Figure 9. The touch screen, orientation sensors and accelerometer are used to detect the gestures

Figure 15: First PC-based prototype

and switch between vertical and horizontal modes. Additional screenshots are in Appendix C.

The video archive consists of a month's worth of 3–5 minute news video clips downloaded from the CNN website. These clips are pre-segmented, indexed by category, and annotated with editor assigned descriptive keywords. The videos are delivered using the standard client-server distribution model, and are in 70kbps MPEG4-AVC format, hosted on a server running RedHat Enterprise Linux 5 and Apache. Details of the video archive are further described in Section 5.1.

CHAPTER 5

CONTENT RECOMMENDATION ALGORITHM

This chapter describes the video database content and representation, the algorithm for recommending news content, and corresponding simulation results and discussion.

5.1 News Video Database

Individual news video clips are available daily on the CNN website [45], spanning various news stories and categories. A crawler and downloading client program was developed to access the CNN website, download and save the news video clips to local disk storage. News video clips were then downloaded daily from the CNN website over the course of a month in 2006. Each video clip presents a single news story, and has a typical viewing duration of three to five minutes. These clips are pre-segmented, indexed by category, and are annotated with editor assigned descriptive keywords.

The database currently comprises approximately 450 video clips and nine hours viewing time. Each video is represented by a CNN news editor-assigned *title*, *date*, *timestamp*, and *category*. The categories are: *world, US, politics, law, business, science, technology, sports, entertainment* and *health*.

5.2 Video Representation

In addition to the CNN editor assigned title and existing date and timestamp, each video clip's content is represented by manually annotated keywords selected from the spoken audio. Five volunteer subjects were shown all the video clips from the entire database in random order. After viewing each video clip, each subject was instructed to assign some keywords representing video's main story and high level content.

The title, keywords and category name are then processed into an unordered bag of words (tokens), with the duplicate words removed. As the number of distinct words

Table 2: CNN video database attribute examples (before processing)

Title	Keywords	Duration	Timestamp	Date	Category
scanning for danger	singapore scanning technology container busiest port security inspection	300	18:58	10/11	technology
like a huge ball of fire	huge fireball new york plane crash terrorism smoke fire explosion hit buzzing flames debris frightening	118	18:10	10/11	US
south korea reacts to nuclear news	south korea reaction north korea nuclear test kim jong il	125	13:25	10/10	world
...	...	...	...	...	...

used to describe a video may be large, duplicate words are removed and the remaining words subsequently undergo stopping and stemming, oft used strategies when setting up information retrieval systems.

Stopping removes "stop words" that have no use in an index and have little direct semantic meaning of the content. For example, words such as "and", "the", "of" are removed. Stemming is commonly used as part of a term normalisation and dimension reduction process. The Porter stemming algorithm (or "Porter stemmer") [46] is the most well known method for removing the more common morphological and inflexional endings from words in English. Words are reduced to their root form (e.g. the words "run", "ran" and "running" all reduce to "run"). These processed words are now terms.

Each video is then represented by this feature vector of terms. A video representation can have any number of distinct terms, and the length of the vector is the number of terms used to represent the video.

5.2.1 Automatic Annotation using ASR

Eventually, the goal is to transition to automated methods of extracting the video semantics, such as using automatic speech recognition (ASR), OCR and text classification to obtain the categories. Currently, audio transcripts can be obtained easily and automatically by using existing available ASR software, without requiring a human to watch every single video and perform manual annotation. Obtaining full audio transcripts or closed captioning provides a larger number of keywords to annotate a video with, enhancing the description of the semantics of the video. The representation is easily extensible to full audio transcripts or closed captioning obtained from the video, and eventually incorporating other attributes such as multiple news categories.

5.3 Naïve Bayes

The Nave Bayes classifier is a core technique in machine learning [47], adapted for areas such as pattern recognition, information retrieval and spam filtering.

A widely used framework for classification is provided by Bayes' Theorem:

$$P(C = c_j | X_1 = x_1, \ldots, X_n = x_n) = \frac{P(C = c_j) P(X_1 = x_1, \ldots, X_n = x_n | C = c_j)}{P(X_1 = x_1, \ldots, X_n = x_n)} \tag{1}$$

Where C is a random variable whose values are one of the classes $(c_1, \ldots, c_j, \ldots, c_e)$, and X is a vector random variable whose values are feature values $(x_1, \ldots, x_j, \ldots, x_n)$. $P(C = c_j | X_1 = x_1, \ldots, X_n = x_n)$ is the conditional probability that a certain feature vector X belongs to the class c_j.

$P(C = c_j | X_1 = x_1, \ldots, X_n = x_n)$ is not known, therefore Bayes' rule suggests estimating $P(X_1 = x_1, \ldots, X_n = x_n | C = c_j)$, $P(C = c_j)$ and $P(X_1 = x_1, \ldots, X_n = x_n | C = c_j)$ and then combining those values. Estimating $P(X_1 = x_1, \ldots, X_n = x_n | C = c_j)$ is also problematic since there are a large number of possible values for each feature in the vector $(x_1, \ldots, x_j, \ldots, x_n)$. Expanding via the product rule shows the large number of parameters that need to be estimated and can only be done if a huge number of training examples were available.

$$P(X_1 = x_1, \ldots, X_n = x_n | C = c_j) = \; P(X_1 = x_1 | C = c_j) P(X_2 = x_2 | X_1 = x_1, C = c_j) \ldots$$
$$P(X_n = x_n | X_1 = x_1, \ldots, X_n = x_n, C = c_j)$$

To simplify, the assumption is that inside each class, the probability of each feature x_j is statistically independent of the occurrence of any other feature x_i. This is the Conditional Independence Assumption. This simplifies it to the product of the

individual probabilities:

$$P(X_1 = x_1, \ldots, X_n = x_n | C = c_j) = \prod_{i=1}^{n} P(X_i = x_i | C = c_j) \tag{2}$$

Substituting this into Equation (1) it becomes:

$$P(C = c_j | X_1 = x_1, \ldots, X_n = x_n) = \frac{P(C = c_j)}{P(X_1 = x_1, \ldots, X_n = x_n)} \prod_{i=1}^{n} P(X_i = x_i | C = c_j) \tag{3}$$

The maximum likelihood estimates for each of the terms can use the frequencies in the data, and then the estimate of $P(C = c_j | X_1 = x_1, \ldots, X_n = x_n)$ can be used to perform classification.

5.4 Naïve Bayes for Recommending News Content

The algorithm is content-based and uses a Naïve Bayesian classifier [47] adapted to handle a "bag of words" representation and a relevance feedback procedure to learn a single user's preferences through explicit voting feedback and then predicting the next video to show to the user.

The task is modelled as a probabilistic binary classification problem of predicting the probability that a new video shown to the user will be voted up, based on set of labelled examples given by user feedback (i.e. the videos that have already been voted upon). A ranked list of potential videos to display is determined, and the next video is selected to be shown to the user.

When a video is voted up, the video is counted as a positive example to the classifier, when voted down, it is taken as a negative example. These labelled examples are used to estimate the probability that a certain new video clip will be relevant based on the observed fractions of words appearing in voted-up and voted down videos. The Naïve assumption of independent words is used to simplify estimates and calculations.

A binomial text model is employed, where each video is represented as a binary vector of distinct terms $X = (x_1, x_2, \ldots, x_n)$, taken from the vocabulary of words V, where $x \in V$ and each x is a binary feature. Class $c_j \in C$. In this case there are two classes c_1 and c_2 where one is the class of voted up videos and the other, voted down videos.

The posterior probability of each class given the term vector X is computed using Bayes' Theorem:

$$P(c_j|X_1 = x_1, \ldots, X_n = x_n) = \frac{P(c_j)P(X_1 = x_1, \ldots, X_n = x_n|c_j)}{P(X_1 = x_1, \ldots, X_n = x_n)} \tag{4}$$

Upon making the conditional independence assumption, where each word occurrence is independent of all other words, we get:

$$P(X_1 = x_1, \ldots, X_n = x_n|c_j) = \prod_{i=1}^{n} P(X_i = x_i|c_j) \tag{5}$$

After assuming conditional independence, Bayes' Theorem is now:

$$P(c_j|X_1 = x_1, \ldots, X_n = x_n) = \frac{P(c_j)}{P(X_1 = x_1, \ldots, X_n = x_n)} \prod_{i=1}^{n} P(X_i = x_i|c_j) \tag{6}$$

The maximum likelihood estimates use the frequencies in the training data, in this case - the number of positive and negative voted examples, the observed fraction of times a word is present in a positive example, and the observed fraction of times a word is present in a negative example.

$$P(c_j) = \frac{N(c_j)}{N} \tag{7}$$

$$P(X_i = x_i|c_j) = \frac{N(X_i = x_i, c_j)}{N(c_j)} \tag{8}$$

The log odds scale was used to avoid the normalizing constant denominator and the implementation issues of multiplying a lot of small numbers together which might

lead to numerical underflow as seen in Equation (9). The logarithms of the probabilities are added instead of multiplying them. Finally, Laplace smoothing is used to avoid zero probability estimates for words that do not appear.

$$\log \frac{P(C_1|X_1 = x_1, \ldots, X_n = x_n)}{P(C_2|X_1 = x_1, \ldots, X_n = x_n)} = \log \frac{P(C_1)}{P(C_2)} + \sum_{i=1}^{n} \log \frac{P(X_i = x_i|C_1)}{P(X_i = x_i|C_2)} \quad (9)$$

Based on the news domain, the time, date and recency matter as breaking and recent news videos are desired by users. Therefore, this posterior probability is computed over a sliding window of news videos ordered by the timestamp from the most recent to the least recent. A subset of most recent news videos are considered first, and as the viewing session progresses and videos are depleted from the database, the next most recent set of videos are considered. After some length of time, older votes are also pruned from the system.

Recent results have shown that Naïve Bayes is less complex, yet relatively efficient and performs competitively when compared to other more complex algorithms [48] and is effective in other recommendation applications [41]. The computational complexity is $O(N)$, linear to the size of the training and testing data. It has been successfully applied in other domains such as email spam filtering with good results [49].

5.5 Information Retrieval Metrics

Precision and recall are the most frequently used measures of an IR system's performance [50]. These are first defined for the simple case where an IR system returns a set of items for a query. These are then extended based on the specific application and retrieval situation.

- Precision (P): The fraction of the number of relevant items retrieved to the total number of irrelevant and relevant items retrieved (in the session)

$$Precision = \frac{\text{number of relevant items retrieved}}{\text{number of retrieved items}}$$

- Recall (R): The fraction of the number of relevant items retrieved to the total number of relevant items in the entire database

$$Recall = \frac{\text{number of relevant items retrieved}}{\text{number of relevant items}}$$

Precision and recall trade off against each other and is usually depicted on a precision-recall curve. A recall of 1 can always be obtained by retrieving all the videos in the database, but the resultant precision will be very low. Recall is a non-decreasing function of the number of videos retrieved. Precision usually decreases as the number of retrieved items is increased.

The measures rely on a collection of videos, and an information need (query) for which the relevancy of the videos in that collection is known. Binary relevancy is assumed - The video is either relevant or not-relevant. In practice however, queries may be ill-posed and/or there may be different shades of relevancy. Other metrics will need to be designed based on the specific application.

5.6 Recommendation Quality Evaluation

Some simulated experiments were conducted to evaluate the semantic quality of the system and determine whether such content recommendation is useful for news videos on a mobile device.

Three representative queries table 3 consisting of two week's worth of video were assembled from the available video clips and their corresponding set of relevance judgments, based on the news story content present in the database. This was achieved by watching every single video clip and determining if it is considered relevant or not, based on the given representative query.

Table 3: Representative queries

Query Set	News Stories & Topics
1	Mark Foley scandal, North Korean nuclear test, plane crashes
2	Political stories, plane crashes, Iraq, Middle East reports
3	Iraq, Afghanistan, other Middle East reports, North Korean nuclear test

Each video set and its relevance set were then used to simulate a viewing and feedback session in the semi-automated content recommendation modality for each representative query. The experiment assumes that each video is viewed only once, and for each video viewed by the simulated user, either an up or down vote is returned. An entire *viewing session* consists of watching a certain number of videos, n in sequence and having voted on each of them.

The classical IR metrics of *precision* and *recall* are adapted for use in this case. The number of videos per session is increased in increments of 10 until the maximum number of videos is reached, to obtain varying degrees of *recall*. Each video retrieval viewing session undergoes 100 trials beginning with a randomised starting video, and the averaged *precision* is computed over all the sessions in the trial.

5.7 Analysis of Content Recommendation Performance

To observe the learning performance given varying amounts of simulated user feedback examples for each representative query, precision is shown against the number of feedback iterations (or videos voted). This is benchmarked against a baseline of purely random retrieval of video clips. The results are illustrated in Figure 16, Figure 17, and Figure 18.

The curves suggest that the current method can produce reasonably effective recommendations for the news video database. The precision improved after around ten feedback iterations; suggesting that after a short time interacting with the system, it will produce relevant videos to the user. However, a longer voting session with more feedback is required to produce consistently relevant videos above the random

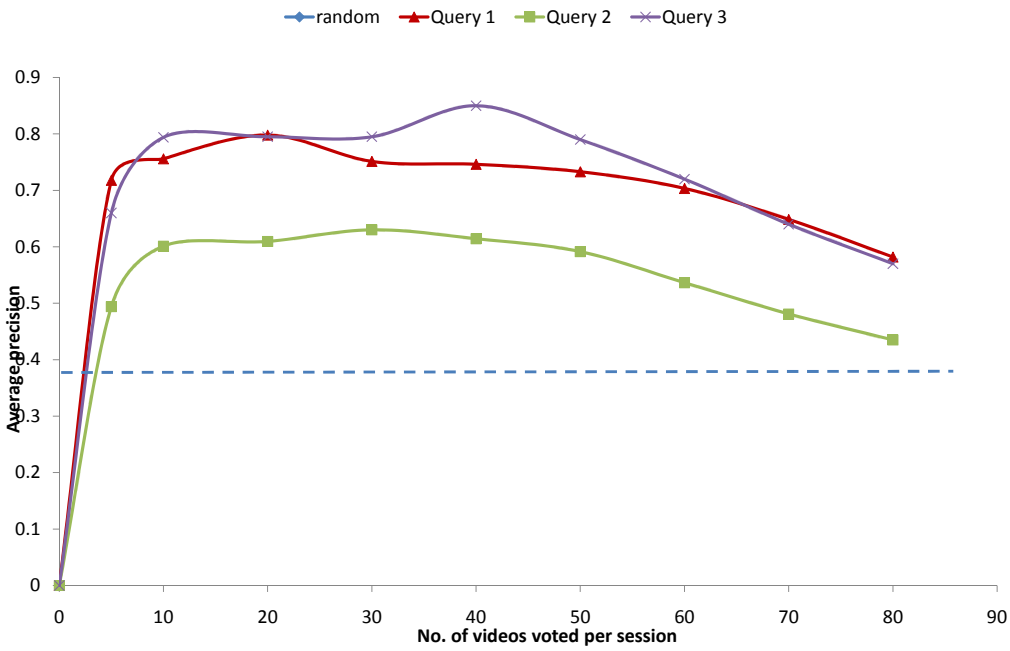


Figure 16: Precision for all 3 representative queries

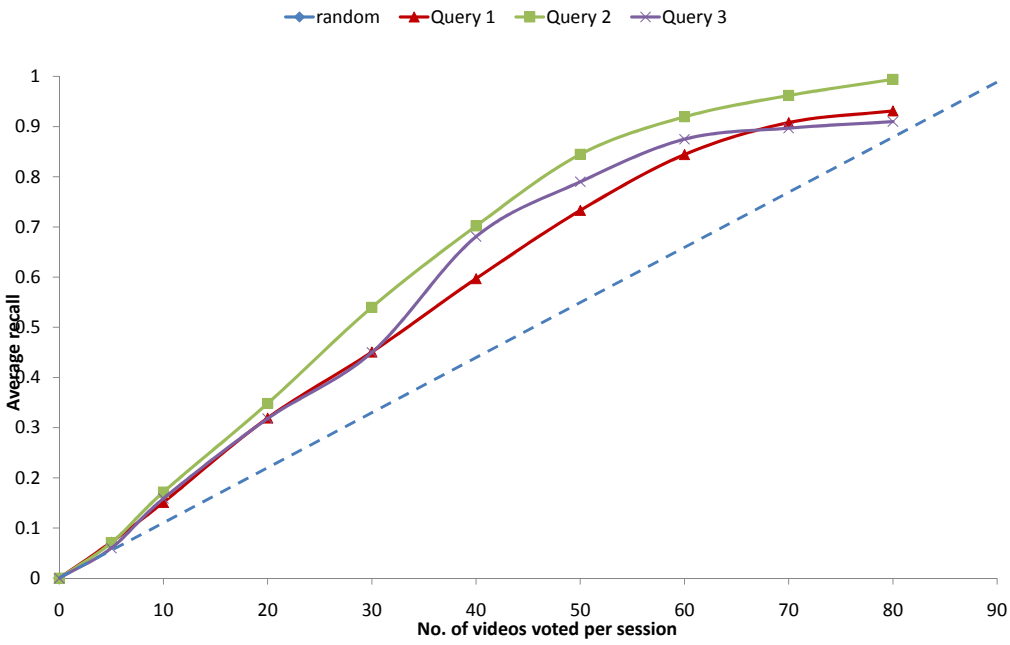


Figure 17: Recall for all 3 representative queries

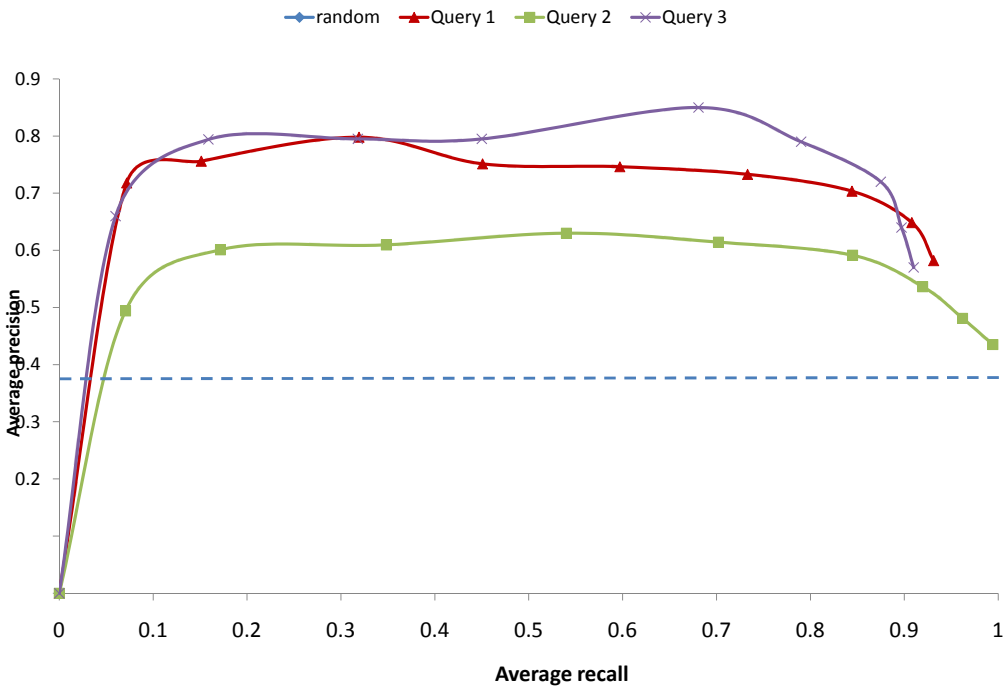


Figure 18: Precision-recall for all 3 representative queries

baseline.

Intuitively, the notion of time taken to transfer and watch a video is proportional to the number of videos seen in a session. It is therefore useful to maximize precision early in the session to reduce the time taken to view videos. As the relevant news stories are exhausted from the database, the precision value decreases. In the case of Set 2, not having many common distinguishing word features results in lower precision accuracy. Further evaluation with larger video test sets, new queries and a large population of actual human users interacting with a full mixed-mode system will be required.

CHAPTER 6

PREFETCHING FOR MIXED-INITIATIVE VIDEO

Caching is the local storing of resources that are expected to be requested again locally. Although it improves user perceived delays, it improves the response times only for cached responses that are subsequently requested. It does not work for newly requested objects, thus the benefits are limited, especially for objects that are viewed once or less popular. To further reduce the retrieval latency, prefetching is the more attractive solution.

Prefetching uses the idle time of the network to retrieve data that the user will likely require in the near future. Prefetching reduces the user access time, but simultaneously, it requires more data to be transferred and increases network traffic. Therefore if the items to be prefetched can be predicted in an accurate manner,

Prefetching and caching are techniques frequently used to reduce user perceived latency.

This chapter discusses video caching and prefetching and contrasts them with the web versions, and the details of incorporating prefetching and client side caching of the video prefixes in this specific mixed initiative system. The various prefetch strategies based on the RF patterns are evaluated with their tradeoffs and the subsequent results.

6.1 Web Prefetching

The concept of prefetching is to make use of the idle periods of the network to prefetch and download data that the user will be likely to request in the near future, so as to reduce network latency. This technique was first developed using the instruction cache to reduce memory latency in microprocessors, and was later adapted to reduce the network latency in accessing web pages [51]. Prefetching and caching methods

have also been adapted for use in various areas such as web proxying and file accesses in operating systems.

In the case of web prefetching, while a user is viewing a retrieved web page, some of the hyperlinked web pages are prefetched in advance according to the user's access history and calculated user access probabilities of these pages [52]. When one of the prefetched pages is indeed requested by the user, the page is available in client memory and can immediately be displayed. Therefore, the user perceives no wait or transmission latency.

Prefetching may reduce user access time, but simultaneously requires more bandwidth and increases network traffic and bandwidth consumption [53]. A significant factor for a prefetching algorithm's ability to reduce latency is deciding which objects to prefetch in advance [54]. A high hit ratio should be achieved, whereby the requested items are successfully found in the set of prefetched items, compared to the total number of requests. On successive requests, in addition to prefetching, the cache replacement policy decides which objects will remain in cache and which are evicted to make space for new items.

Web prefetching is not quite widely implemented in practice, as it is not a trivial task to accurately gather user intent and predict user access patterns to web pages. This user access pattern varies widely depending on time and user population. However, prefetching can be adapted for use in a relevance feedback system as the prediction difficulties are mitigated in this environment. For web accesses, the user exclusively determines the next page to request.

On the other hand, for relevance feedback in information retrieval, the multimedia items are determined by the retrieval algorithm [55]. Given the user's voting feedback and the system's knowledge of videos that might be relevant to the user, the items to prefetch can be predicted with reasonable accuracy. Prefetching can then performed while the user is viewing the current media item, to improve the user perceived latency.

6.2 Video Caching

In common web configurations, the proxy server exists between clients and web servers. The proxy server intercepts the requests from clients, and serves the requested object if it is present in the proxy. Otherwise, it retrieves new objects from the appropriate web server, then caches the new objects or updates the existing objects if it has been modified since the last reference. This reduces the client-end perceived latency.

The existing techniques for caching text and images are not appropriate for streaming media objects as their access pattern differs from web pages. Additionally, the large sizes of typical media objects makes regular web caching strategies difficult, as they quickly exhaust the available cache space. Therefore, the strategies of *prefix caching* and *segment-based caching* are employed for large streaming media items such as videos.

Prefix caching stores only the beginning of a video, and minimizes the storage requirements. It performs well when most clients access the initial portions of media objects [56]. It also reduces the startup buffering delay by immediately serving the cached prefix to the client while retrieving the subsequent video segments from the original server. Segment-based or partial caching provide added flexibility by caching segments, rather the media objects in their entirety.

Streaming video proxy caching systems already use prefetching techniques, prefix and segment-based caching to minimize the playback startup latency [57, 58]. All these require accurate prediction of the client access pattern for best results and are tuned towards the access characteristics of the media objects. Most work concentrate on predicting the frequency and popularity of media items in a generalized video-on-demand or web video access environment, rather than looking at the exact content of the video.

6.3 Prefetching Strategy

As discussed in Section 2.5, streaming video involves a startup delay to fill the play-out buffer. While this buffer can compensate for the delay jitter, it introduces an additional user perceived delay on the client end, and is significantly long at lower bandwidths. User experience for the mixed-initiative system thus may be unsatisfactory despite high semantic quality when relevant videos are being delivered, when there is a delay for every video viewed in sequence.

The usage access pattern is also known in a relevance feedback IR context, and therefore the subsequent items to be viewed can be predicted. As demonstrated in [55] for a content based image retrieval (CBIR) system, the images can be predicted with reasonable accuracy by the retrieval algorithm to achieve lower user perceived latencies.

As an extension of prefetching for CBIR and adapting the methods used in network video caching and prefetching, it is possible to perform prefetching for streaming video in an IR relevance feedback environment. The idea is to intelligently prefetch the prefixes of the video clips to local cache, reduce the latency observed at the client side and transfer some of the data load from server to local storage. Given the user's feedback and the system's knowledge of videos that might be relevant to the user, prefetching the more desirable video prefixes can performed while the user is viewing the current video.

The prefetching mechanism is illustrated in Figure 19. To prefetch at iteration n, a set of possible videos for iteration $n+1$ must be predicted before the user feedback at n occurs. The ranked set of videos to prefetch at iteration n can be denoted as the set $R_n = f(V_{n-1}, H_{n-1})$, where $V_n - 1$ is set of voted videos up to iteration $n-1$, H_{n-1} is the set of votes in the voting history up to iteration $n-1$, and f is the content recommendation algorithm given in Equation (3), or any other relevance feedback content recommendation method or similarity measure.

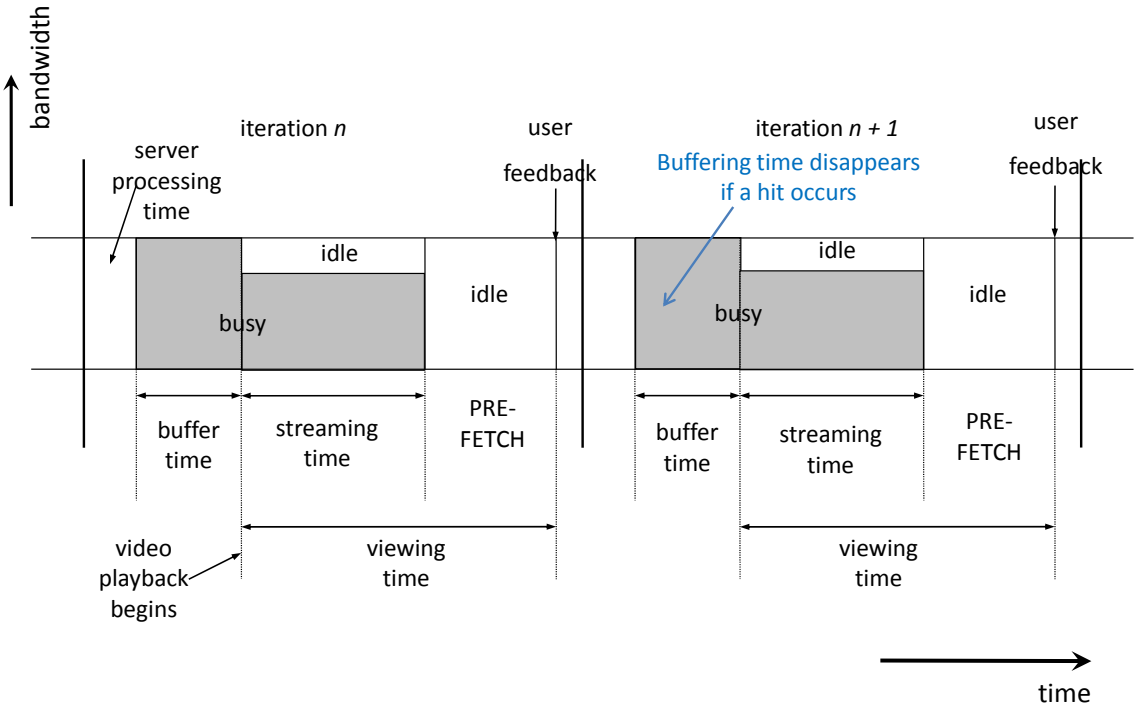


Figure 19: Prefetching scheme

From Figure 19, an iteration can be thought to comprise of 3 main stages, *buffering*, *streaming* and *fully idle*. Prefetching begins during the streaming (video watching) stage, using any leftover bandwidth that is not used to stream the currently viewed video. The feedback vote from the user may arrive at any time during the viewing period, either in the streaming stage or fully idle stage. When a vote-down is received during the streaming stage, the fully idle stage is never reached as the system advances immediately to the next video. The number of video prefixes prefetched may be limited. When a vote-up is received during the streaming stage, there is an opportunity to recompute the ranking $R_{n+1} = f(V_n, H_n)$ on this new feedback on the current iteration n and repopulate the prefetch buffer, as it is likely the user will continue watching the video till completion.

6.3.1 Buffer Replacement

To selectively prefetch and reduce repeated transfer of possibly the same items to fill the client side buffer slots, the *prefetch buffer replacement scheme* would essentially be obtaining the ranked list of videos from the server R_n using Equation (9), identifying the videos that are currently not residing in the client side buffer slots based on the given ranked list, and populating those videos to the buffer.

In set theory notation, $\setminus$ is the relative complement (i.e. set-theoretic difference). Where R_n denotes the set of ranked videos from the server at current iteration n, and B_{n-1} is the set of items in the buffer from the previous prefetch cycle $n-1$, the set of videos to add, V_{add} is:

$$V_{add} = R_n \setminus B_{n-1}$$

The videos to evict from the prefetch buffer is the set $B_{n-1} \setminus R_n$, and the set of videos that remain is $B \cap R$, resulting in a reduction of actual bytes that are transferred over the network.

6.3.2 Prefetch Buffer Selection Scheme

In this environment, the prefetched items have a relative content relevancy ranking as opposed to conventional prefetching. This information is useful to implement differing strategies:

Prefetch by best semantic quality: If semantic relevancy is deemed more important, the video should be fetched even if it does not appear in the prefetch buffer. There is a miss and the cost is in the form of user perceived latency when time is taken to fetch the video and buffer it before streaming.

Prefetch by best perceived latency: If bandwidth or time is costly, the video should be selected from among the prefetched items in the buffer slots, even if it is not entirely optimal. This method achieves a 100% hit rate at the cost of semantic

quality. In this case, the highest ranking video in the prefetch buffer is selected. Therefore, a tradeoff between the semantic quality and the latency incurred needs to be considered.

6.4 Prefetching Metrics

Performance measurements of prefetching techniques are primarily in terms of *hit ratio* and *bandwidth usage* or wastage. Some of the metrics are summarized below:

Session Hit Ratio: the ratio of requests that are serviced from prefetch buffer, to the total number of requests in the session.

Bandwidth Usage: the number of bytes that are used for prefetching the video prefixes during the entire session

User Perceived Latency: the user observed latency from the prefetch technique in the entire session, i.e. the sum of buffering times when there is a miss.

The *precision* and *hit ratio* represents the performance and guessing ability of the prefetch strategy. As the idea of prefetching is to reduce the client side latency, the average user perceived latency for the entire session is measured.

6.5 Analysis of Prefetching

A simulation was done to evaluate the performance of the prefetching strategies. Performance measurements of regular prefetching techniques are primarily in terms of hit ratio and bandwidth usage or wastage. As the semantic quality is important, precision is also used to measure the relevancy of the output.

Based on the representative queries detailed in Table 3, the video data sets and their relevance sets were used to simulate multiple viewing and feedback sessions of 20 video viewing iterations in both the the semi-automated and automatic random modes. Each session undergoes 50 trials beginning with a randomised starting video, and each metric is computed over all the sessions in the simulation. Each buffer slot

size was fixed at 5s worth of video, and there is a limit to the number of buffer slots (i.e. number of video prefixes prefetched). The viewing time for a voted down video was 20s, and a voted up video was viewed for the entire playback duration.

To observe the reduction in latency from prefetching, *prefetching by best semantic quality* is employed with varying buffer slot sizes and between the two modes of semi-automatic and automatic with random video selection. As seen from Figure 20, the recommendation algorithm offers accurate prediction for prefetching and latency reduction, increasing as the buffer slot size is increased. There was a trade-off between storage capacity usage and the observed latency benefits. Figure 21 demonstrates that filling the prefetching cache randomly does not offer much benefit, as there is no prediction involved.

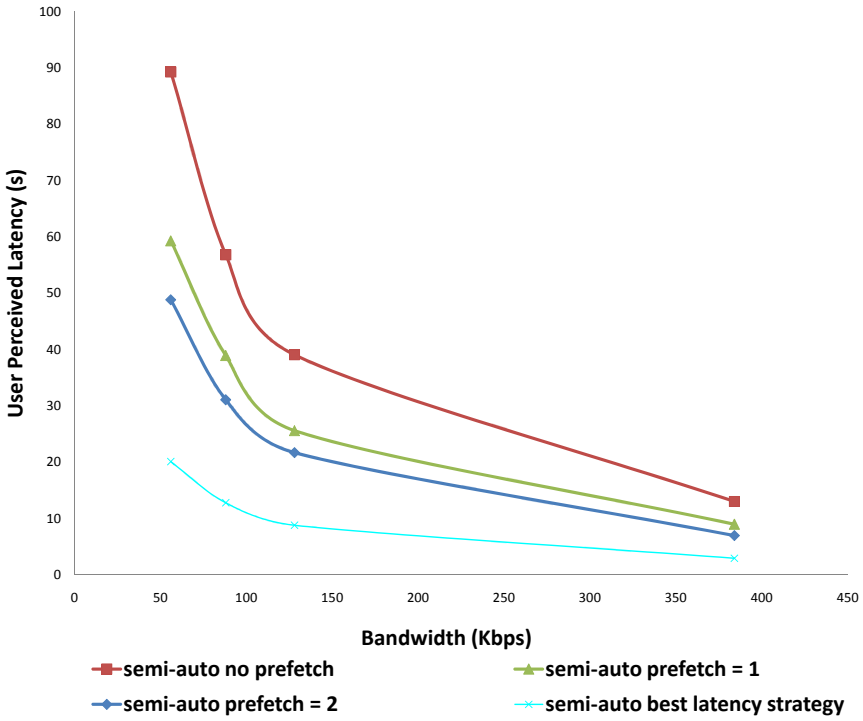


Figure 20: Prefetching Latency Reduction

Overall, the latency reduction is almost constant with the bandwidth. Prefetching by best perceived latency gives the lowest user observed wait time, but it is shown

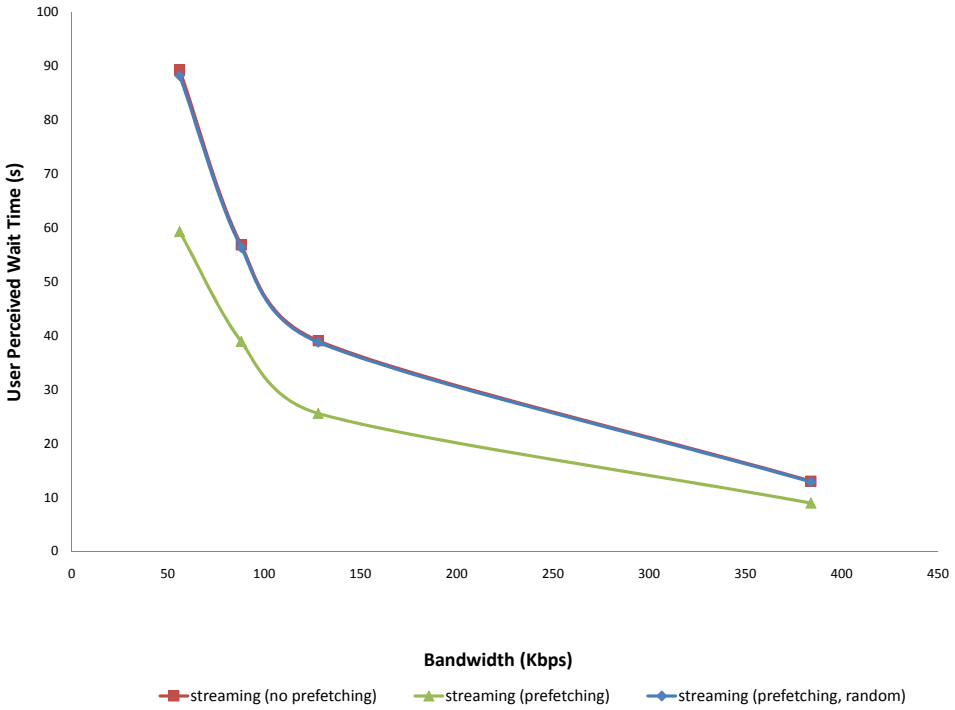


Figure 21: Random Prefetching Latency Reduction

later it is at the expense of precision in Figure 23. Figure 22 suggests that if there was sufficient storage, prefetching larger numbers of video prefixes would generally result in a higher hit rate, and thus reduce the user perceived latency.

Finally, the tradeoff between semantic relevancy and latency was demonstrated when prefetching by best perceived latency. In this experiment, a given maximum threshold of perceived latency is allowed for a session, with a given limited number of prefetch slots. As seen from Figure 23, limiting the time allowed forces the fetching of less optimal results from the prefetch cache. Thus, there is a tradeoff between semantic quality and latency if time or bandwidth is to be conserved.

The results illustrate overall that if prefetching is performed in a reasonably intelligent manner, the user perceived latency can be reduced without excessive bandwidth or client storage usage, and without incurring a large reduction in the number of relevant videos viewed.

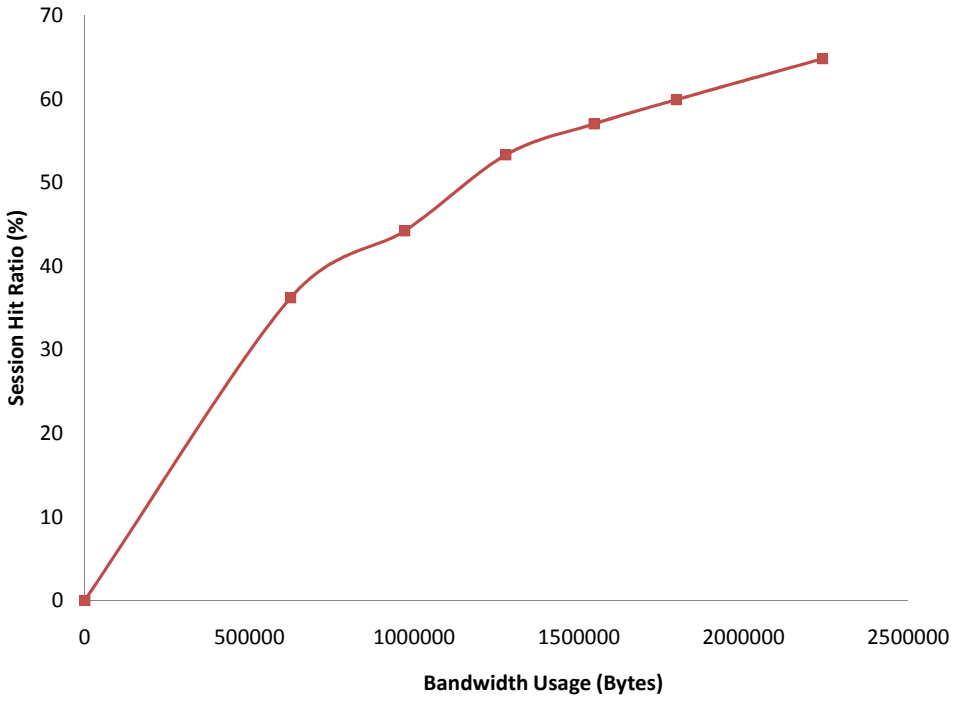


Figure 22: Session Hit Ratio vs. Bandwidth Usage

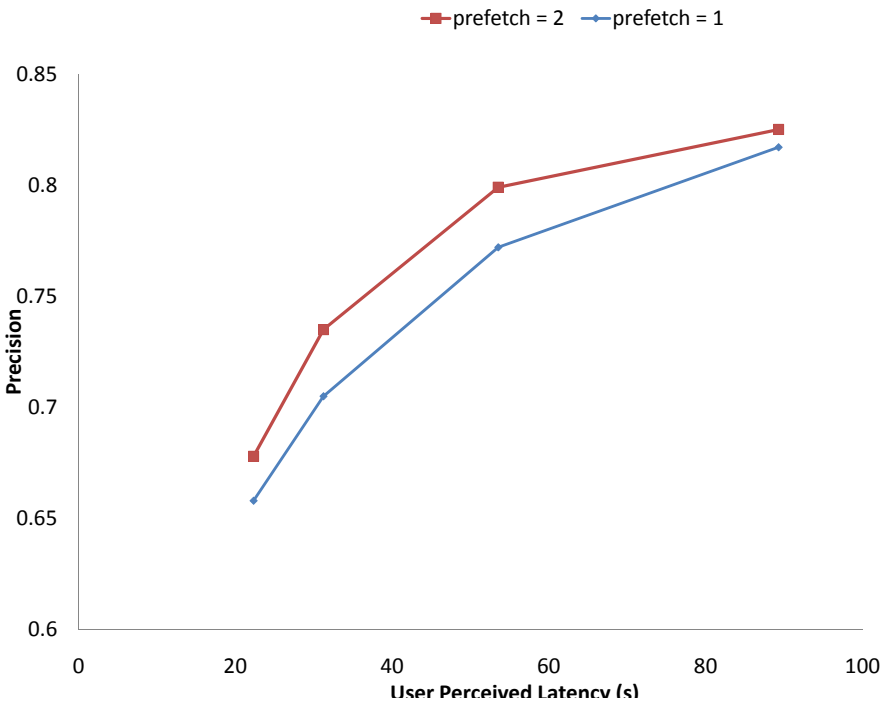


Figure 23: Precision vs. User Perceived Latency

CHAPTER 7

USER EVALUATION

To evaluate the performance of the mixed-initiative system, a user experiment with the CNN video database and a set of human volunteers interacting with an actual mobile device was conducted. The benchmark is a random presentation of the video clips in automatic mode on the mobile device, and standard video streaming with a buffering delay before playback. These benchmarks were compared against the proposed mixed-initiative recommendation method, and streaming with prefetching of the video prefixes for lower user perceived latency.

7.1 Experimental Variables and Hypotheses

The independent variables of the study are the different news content recommendation methods, which include an automatic random presentation of the news video clips, and semi-automatic content recommendation using Naïve Bayes. The other variable is the prefetching method, namely standard video streaming with a buffering delay before playback (no prefetching invoked), and streaming with prefetching of the video prefixes.

The dependent variable is the user evaluation of the systems which include objective performance measures, and subjective user satisfaction measurements based on user surveys, further described in Section 7.3.

The user experiment aims to discover whether the proposed news recommendation method performs better than a simple random presentation of the day's news video clips, and also whether video prefix prefetching together with the proposed content recommendation improves user perceived latency, based on the given objective and subjective measures listed in Section 7.3.

The user experiment was designed as three sets of matched viewing session pairs:

1. **Automatic random (no prefetching) vs. semi-auto content recommendation Naïve Bayes (no prefetching)**

   Long waiting times experienced by users in both viewing sessions, due to the simulated buffering before playback of videos.

   Hypothesis: Users prefer semi-auto recommended content over random presentation of videos as more relevant results are presented, despite the perceived latency.

2. **Semi-auto content recommendation Naïve Bayes (no prefetching) vs. semi-auto content recommendation Naïve Bayes (with prefetching)**

   Longer user perceived latency for the most semantically relevant results. Shorter wait with cache set to prefetch by best perceived latency, but less desirable and more irrelevant results are presented.

   Hypothesis: Users prefer to wait longer for most relevant results on semi-auto Naïve Bayes.

3. **Semi-auto content recommendation Naïve Bayes (no prefetching) vs. Automatic random (with prefetching)**

   Longer user perceived latency for relevant semi-auto Naïve Bayes results, but a relatively short wait for random results.

   Hypothesis: Users still prefer to wait even longer for most relevant results on semi-auto Naïve Bayes, rather than be given a quick random presentation of videos they may not like.

7.2 Experimental Setup

The user experiments were conducted using a specially designed Google Android mobile phone application that displays the video clips and accepts user input. The

system displays video clips in a sequence to the user in an automatic random mode or the semi-automatic content recommendation mode with Naïve Bayes, and simulates varying delays for buffering delay and prefetching.

The user is requested to watch each video and respond to the system based on the specific usage task (Representative query task 3 described in Table 3) using the experimental system. In the case of the semi-automatic content recommendation mode, the user relevance feedback vote was collected, and the next best video clip was then computed and displayed to the user. For the automatic random modality, a pseudo-random sequence of video clips is displayed to each subject to ensure consistency. The same random sequence of video clips was given for each experiment. The user feedback vote was collected, however the votes did not affect the subsequent video in the random sequence.

In all cases, all user activity such as the up or down votes, identity of the viewed video clip, and length of time spent viewing the video were logged to file and monitored on the Google Android mobile phone. The mobile client application was loaded on two similar devices: the HTC Dream (T-Mobile G1) and HTC Magic (MyTouch) for the purposes of the user experiment. Both devices have the same screen resolution, processor and similar physical dimensions.

Paper-based and online web survey forms were developed to assess user satisfaction and collect user feedback and comments after the viewing sessions and experiment sets. The paper-based version of the forms were used if the subjects were unable to access the web or had limited network access. The online web version of the forms were coded in HTML and PHP, and hosted on a server running RedHat Enterprise Linux 5 (RHEL) and Apache. When the user submits the form online, the survey results are processed and stored in a mySQL database on the same server. Examples of both the paper and online web-based user forms can be seen in Appendix B.

Table 4: Viewing session survey 5-point Likert scale questions

Statement	Satisfaction Measurement
It is easy to use this system	UI ease of use
The wrong videos are being shown to me	Accuracy/Content Relevance
It took too long for a video to load	User Perceived Latency

Table 5: Exit survey questions

Question
Which session did you prefer?
Why did you prefer this session?
Any other comments?

7.3 Measurement Instruments

Both objective and subjective measures were used to assess user satisfaction of the subjects in the experiment. For the objective measure, the classical metric of *precision* is used, as described in Section 5.5.

The subjective measures include a post-viewing survey, assessing user satisfaction pertaining to that specific viewing session with regards to the UI ease of use, content relevance and perceived latency. All question responses are measured on a 5-point Likert scale with 1 being "Strongly Disagree" and 5 being "Strongly Agree". The questions are detailed in table 4. There is also a free response question that allows the user to enter any other additional comments. This survey is given after the completion of each viewing session.

Finally, a post-experiment exit survey was also administered after every experiment set, asking the user to choose the preferred viewing session out of the two given, and to state the reason for the preference (Table 5). There was also a free response question for any other additional comments. The details of the user survey forms are listed in Appendix B.

7.4 Experimental Design & Procedure

Sixteen volunteer subjects were recruited for the viewing experiments using the CNN video clip data set. They were between the ages of 22 to 31, and mostly consisted of graduate students or young working professionals. Most reported that they were comfortable using technology and smart mobile devices, although only half of them owned an actual mobile device or "smartphone" capable of video playback.

There were three sets of experiments comprising of a pair of matched viewing sessions for each set. For each subject, these three experiment sets were conducted in succession. Each set consisted of two viewing sessions, one session is the control and the other with the dependent variable changed, used for comparison. The ordering of both the experiment set and viewing sessions was randomly assigned to each subject, and the tests were administered blind. The subject was not informed beforehand regarding the type of test condition that will be administered, or the specific experiment set that will be conducted. Each viewing session length was limited to a maximum of 15 mins, and maximum 30 mins per experiment set with 5 minutes rest in between sessions.

At the beginning of the experiment, the subject was given a form with instructions and an explanation of the purpose of the experiments. These user instruction forms are also listed in Appendix B. The experiment, mobile UI and usage of the application and mobile device was also verbally explained to the subject. Time was then allocated to answer any questions that may arise.

For each viewing session, each subject was then requested to vote and respond to the system with the assumption that they are interested in news stories from the given representative query. Each video was viewed only once, and for each video viewed by the subject, a vote has to be returned, either up or down. An entire viewing session was considered complete when 15 videos were watched in sequence. During the experiment, the subject was also encouraged to articulate any thoughts aloud,

Table 6: User experiment precision values

(a) Experiment Set 1 Precision

Viewing Session	Mean	Standard Deviation
Random (no prefetch)	0.314	0.108
Naïve Bayes (no prefetch)	0.747	0.143

(b) Experiment Set 2 Precision

Viewing Session	Mean	Standard Deviation
Naïve Bayes (no prefetch)	0.801	0.108
Naïve Bayes (prefetch)	0.841	0.129

(c) Experiment Set 3 Precision

Viewing Session	Mean	Standard Deviation
Naïve Bayes (no prefetch)	0.761	0.156
Random (prefetch)	0.447	0.074

and was informed that researchers would be observing and taking notes during the entire experiment.

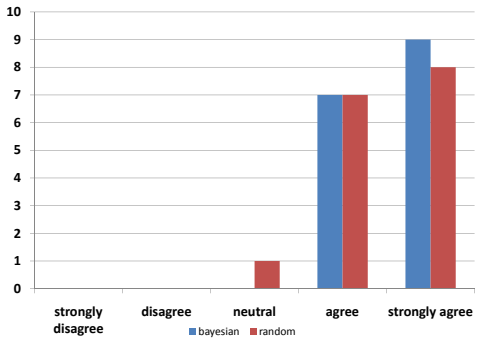
The Likert scale post-viewing survey questionnaire is presented at the end of each viewing session to gather user satisfaction feedback. At the conclusion of each experiment set, an exit survey is then presented, for the user to choose which viewing session out of the 2 was preferred. Finally, an informal verbal interview was conducted to collect feedback and allowing the subject to articulate any matters that were not reflected on the forms.

7.5 Experimental Results & Discussion

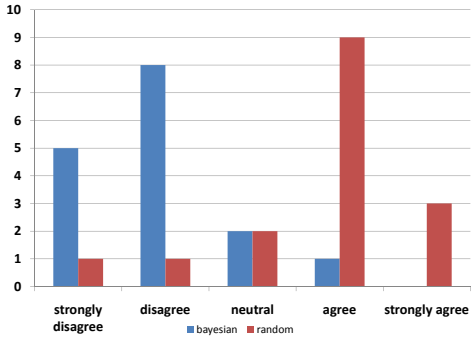
The data collected from the experiments include the number of up and down votes of the subject during each viewing session. These are used to compute the *precision*. Table 6 describes the mean precision values for the experiment sets.

The higher precision values for experiments 1 and 3 indicate that the subjects considered the video clip recommendations from the proposed semi-auto Naïve Bayes method preferable, as they provided more relevant results to the users. As prefetching by best latency method was used for experiment set 2, there are less relevant

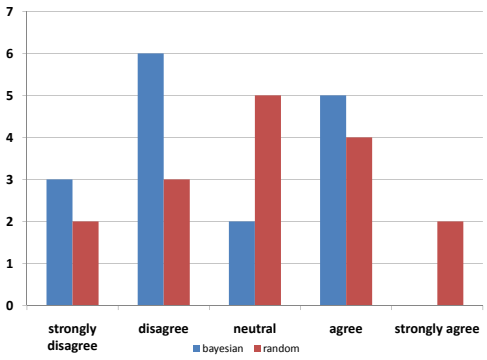
videos presented, due to fetching from the cache, but shorter user perceived latency. However, similar precision values were reported by the subjects in experiment 2. The users did not seem to notice the less relevant results due to prefetching in this specific representative query, or the faster loading of videos might have influenced their voting behaviour.



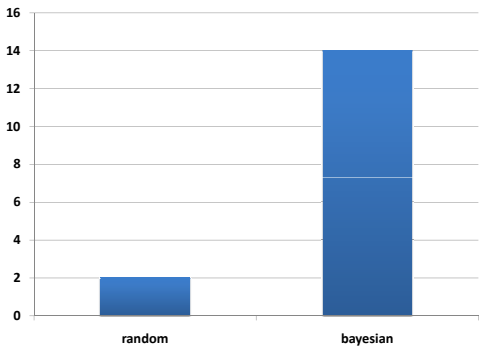
1. "It is easy to use this system"



2. "The wrong videos are being shown to me"



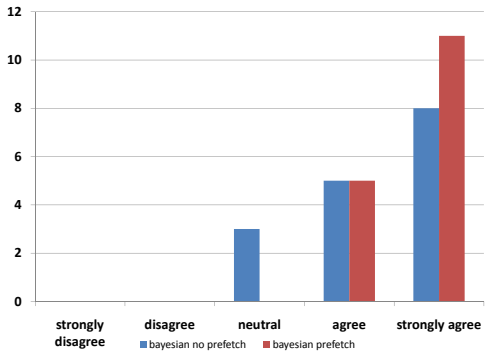
3. "It took too long for a video to load"



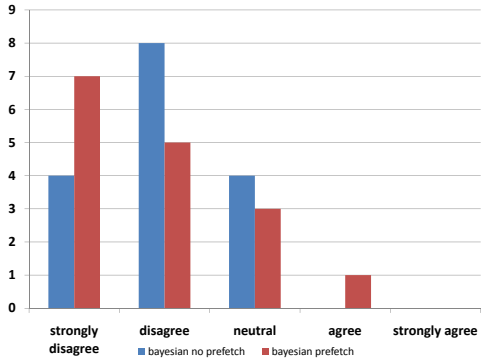
Which session did you prefer?

Figure 24: User experiment 1 survey results

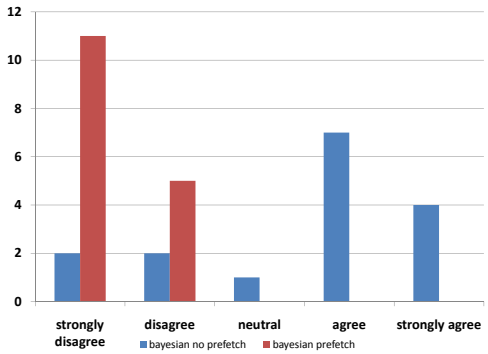
The distribution of user responses regarding each viewing session and their session preference are detailed in Figure 24, Figure 25 and Figure 26. As the Likert scale gives non-parametric data that may not be normally distributed, the Wilcoxon Signed Rank Test is used to measure the performance difference between these survey replies. The Wilcoxon Signed Rank Test can be considered analogous to the paired t-test for



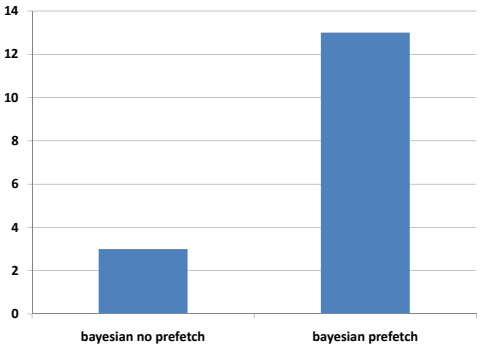
1. "It is easy to use this system"



2. "The wrong videos are being shown to me"



3. "It took too long for a video to load"

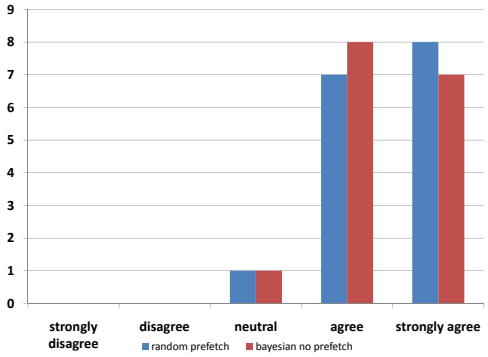


Which session did you prefer?

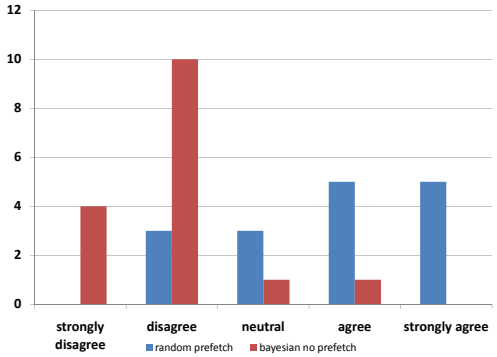
Figure 25: User experiment 2 survey results

continuous data.

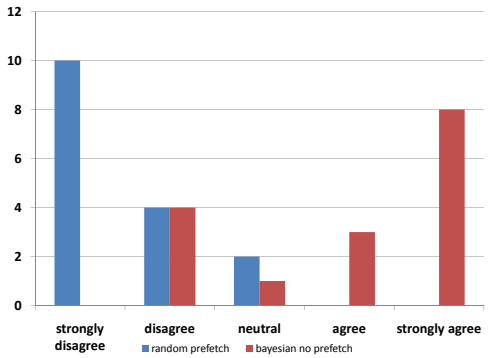
For experiment 1, 87.5% (14 total) of the users indicated a preference for the semi-auto Naïve Bayes method over a random presentation of the video clips. There is a statistically significant response to the survey statement regarding the content relevance ($p < 0.02$), indicating the users noticed a difference in relevant videos being shown in the session. This is also supported by the high precision value of 0.747. As there was no significant change in the other two questions, the users found the UI overall easy to use, and the change in video relevance did not significantly affect user perception of latency. Most users reported seeing more relevant results to what they wanted to watch in the verbal interviews.



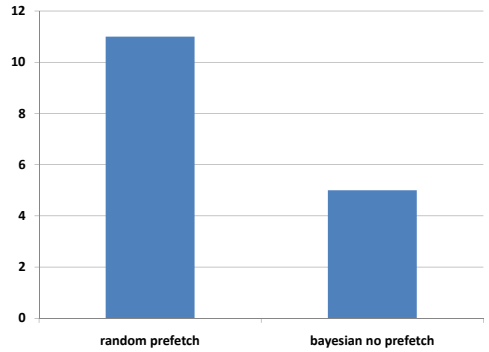
1. "It is easy to use this system"



2. "The wrong videos are being shown to me"



3. "It took too long for a video to load"



Which session did you prefer?

Figure 26: User experiment 3 survey results

In experiment 2, 81.3% (13 total) of the users preferred semi-auto Naïve Bayes method with prefetching enabled, and noticed reduced latency when watching the video clips. This is reflected in the significant W value to the third statement ($p < 0.02$). Based on the similar precision values, the users were not able to tell the difference in relevancy due to prefetching from the cache. Almost all users did not complain of the slightly less relevant results, but instead commented on the faster loading of videos. This was contrary to the hypothesised outcome that users would notice the irrelevant results and indicates speed is of importance to the users. Again, the users found the UI easy to use despite the changes in the other variables.

Experiment 3 had 68.7% (11 total) of the users preferring a random display of video

Table 7: Wilcoxon Signed Rank Test on user experiment sets

(a) Wilcoxon Signed Rank Test for experiment set 1

Statement	W	p-value
It is easy to use this system	1	1
The wrong videos are being shown to me	**-94**	**0.0012**
It took too long for a video to load	-21	0.1484

(b) Wilcoxon Signed Rank Test for experiment set 2

Statement	W	p-value
It is easy to use this system	10	0.1250
The wrong videos are being shown to me	-5	0.6875
It took too long for a video to load	**-91**	**0.0002**

(c) Wilcoxon Signed Rank Test for experiment set 3

Statement	W	p-value
It is easy to use this system	-3	0.8125
The wrong videos are being shown to me	**-91**	**0.0002**
It took too long for a video to load	**91**	**0.0002**

clips with prefetching enabled, than the proposed semi-auto Naïve Bayes method. The signifiant change in survey statements ($p < 0.02$) regarding video loading times and relevant video content indicated the users noticed this change, but tended towards choosing the random viewing session as the preferred one. This also opposed the hypothesis that users would prefer more relevant results at the expense of perceived latency. Based on the verbal interviews, most users felt conflicted between choosing which session was better, but leaned towards faster perceived latency. The users offered the explanation that "If the videos load fast, I can quickly flip to what I want to watch". However, a vocal number of other subjects mentioned that they would soon tire of skipping to the next video after seeing a certain number of irrelevant videos, and would prefer some automated recommendation or filtering to aid the process.

Therefore, it can be safely concluded that the semi-auto recommendation system is preferred over a simple random presentation of the videos for experiment set 1. Additionally, despite slightly less actual precision, prefetching combined with the

semi-auto recommendation system is still preferable than none in experiment set 2. The results are not very clear for experiment 3 and will require further testing to indicate the sweet spot between the relevancy and latency tradeoff. Further details of the Likert scale user survey are depicted in Appendix D.

Additional observations were also recorded, based on the free-response sections of the survey forms and informal verbal interview at the end of the experiments.

For experiment set 1, some users felt the random presentation of videos took longer. However, the simulated buffering delay remained constant for this experiment for both random and semi-auto Naïve bayesian content recommendation. An explanation might be that the high number of irrelevant videos made users pay attention to the waiting time.

In experiment set 2, the majority of users did not notice the larger number of irrelevant videos (lower precision) due to prefetching. The users indicated they were more concerned with the delay. However, some users still preferred to wait for more relevant videos. Despite the constant bitrate and video quality, users complained of artifacts, blockiness and other encoding irregularities during the viewing session with the longer latency. This might be due to a longer waiting time that causes users to pay attention to the video's imperfections and other issues.

Watching the subjects during the experiments also yielded some other interesting observations. A majority of users kept turning the phone horizontally and further maximizing the video to fill the entire screen. Some others asked for headphones to be able to hear the audio better. This suggests that video viewing is a solitary activity on a mobile device and requires higher attention.

CHAPTER 8

CONCLUSION

This dissertation addresses the apparently disparate issues pertaining to multimedia display and access on a mobile device, in the specific context of viewing news video clips. As most existing work consider the components of mobile UI, content retrieval and efficient network transfer separately, an argument is made for a more unified system design, where the coordinated design of these components will provide an effective solution to overcome mobile device constraints.

A mixed-initiative mobile UI that presents the news video clips temporally using voting operations and gestures is proposed as the user-facing component of the system, to overcome mobile input and size limitations. Supporting this mobile UI is a modified Naïve Bayesian recommendation algorithm that filters the relevant videos based on the user's RF. When limited to the news domain, the algorithm does not need to be overly complex and performs competitively, especially when given a window of recent and breaking news items to work with. As evidenced by the user experiment results, users have indicated that such an application is easy to use, and desirable for watching news video clips on a mobile device.

The recommendation algorithm can be possibly be extended to other types of databases such as movies or user generated videos with the inclusion of other features such as movie genre, actor name, or type. The requirement for a sliding window of recent videos or pruning of user votes can be removed, as user preferences in these domains are more invariant. Further enhancement to the content recommendation algorithm is discussed in the subsequent section.

Interactivity is also hampered due to higher perceived latency when accessing media items over the network. The video streaming mechanism and access pattern are characterised using system models and a prefetch method is proposed for the video

prefixes based on the RF pattern, to further eliminate the buffering time. Different prefetch strategies can be employed depending on available bandwidth and user tolerance of waiting time versus content relevancy. The experiments also suggest that a slight sacrifice of semantic relevancy and additional bandwidth may reduce the user perceived latency and improve the interactivity of the system.

As IR is a large multidisciplinary field with many difficult problems, system models have been used for simplicity and focus on the important parameters. The recommendations and conclusions are qualified to the extent they are specific to these models. Overall, the work has given some insight into the various issues affecting mobile access to multimedia, and the possible combined application of methods from the areas of content-based IR, user interaction, caching, and prefetching.

Possible areas of future investigation are described in Section 8.1.

8.1 Further Research

Currently, the content recommendation algorithm only considers recommendations for a single user based on the RF, as the viewing and voting data from a large user population is unavailable. It is desirable to collect this viewing and voting data from a large population based on the news videos and eventually incorporate global feedback, creating a hybrid content-based and collaborative filtering recommendation algorithm for better results. As mobility also brings continuous change in user context and location, this information should also be used to enhance recommendations.

Based on existing studies of user behaviour, users are reluctant to interact more than necessary with the mobile device. Thus, implicit feedback such as the video viewing duration should be incorporated into the algorithm to further enhance recommendations, and additional on-screen gestures can be created. As a complement to this, specific user experiments should be designed to quantify user attention and interest during video clip watching. From the user behaviour standpoint, more work

is also needed to understand how semantic relevancy affects and bias users' perception of video and audio quality acceptability.

Finally, from the system architecture perspective, the content recommendation algorithm should be extended to perform distributed retrieval and prefetching to local networks. Video proxy caching can be performed, instead of caching directly on client which might strain mobile resources. Apart from video prefixes, content based segmentation and caching is also possible for each video clip.

8.2 Contributions

There are three main contributions of this research. The first is the design of an easy to use mixed-initiative UI with an "interactive TV-channel" metaphor for the viewing of news video clips on a mobile device. News browsing and video watching behaviours are first identified through a literature review, and an analysis on how they apply in the context of mobile devices with limited resources. Based on this study, the UI presents a full screen news video and simplified operations to vote up or vote down a video to customise subsequent video clip and improve semantics. Gestures corresponding to each operation are also implemented for the easy navigation and browsing without losing screen estate.

The second is the coordinated design of a content recommendation and RF algorithm and a prefetching method that is suitable for news video clips in such a mixed-initiative environment. The Naïve Bayesian algorithm recommends the subsequent videos based on the user's RF. The prefetching of video prefixes based on the system's knowledge of relevant items then reduces the user perceived latency on the client side.

Finally, there is the user study and evaluation of such a mixed-initiative UI on a mobile device, and overall mobile news watching behaviour observation. The user experiment suggests that users desire specific mobile centric interfaces to assist news

video browsing. In the interest of interactivity and reduced perceived latency, users were willing to sacrifice some relevancy in the recommendations to achieve this. An additional contribution is also the creation of a pre-segmented and annotated CNN news video clip archive for future research use.

APPENDIX A

ALGORITHM PSEUDOCODE

The following is a detailed pseudocode implementation of the Naïve Bayes content recommendation algorithm. The pseudocode is similar in syntax to the Java implementation with some liberties taken for reading clarity.

Naïve Bayes implementation

```
1  // compute the probability score for a given video
2  computeScore(video) {
3
4     prob = 0; // initialize the value to return
5
6     // get the keywords belonging to this video
7     videoKeywordSet = video.getKeywords();
8
9     // for each term in the voted training set
10    for (term : scoresMap.keySet()) {
11
12       // numerator and denominator
13       numer = 0;
14       denom = 0;
15
16       // get the existing up and down votes for this specific↘
            → term
17       scores = scoresMap.get(term);
```

```
18
19          if (videoKeywordSet.contains(term)) {
20             // video contains the term in the training set
21             numer = ((WEIGHT + scores.up) /
22                    (WEIGHT + num_upvotes));
23             denom = ((WEIGHT + scores.down) /
24                      (WEIGHT + num_downvotes));
25          }
26          else {
27             // video does not contain the term in the training ↘
                →set
28             numer = ((WEIGHT + (num_upvotes - scores.up)) /
29                    (WEIGHT + num_upvotes));
30             denom = ((WEIGHT + (num_downvotes - scores.down)) /
31                      (WEIGHT + num_downvotes));
32          }
33
34          prob += Math.log10(numer / denom);
35       }
36
37       totalvotes = num_upvotes + num_downvotes;
38       prob += Math.log10((((WEIGHT + num_upvotes) /
39                            (WEIGHT + totalvotes)) /
40                           ((WEIGHT + num_downvotes) /
41                            (WEIGHT + totalvotes))));
42
43       return Math.pow(10, prob);
44    }
```

APPENDIX B

USER EXPERIMENT FORMS

(This page intentionally left blank.)

Experiment Guide & Instructions

Welcome & Introduction

Thank you for volunteering to participate in this experiment.

Today we are asking you to serve as an evaluator of a mobile news viewing system and to complete 3 sets of news viewing scenarios. The goal is to find out how good this news recommendation system is. We will record your reactions, opinions and clicks, and we may ask you to clarify statements that you make from time to time. Each participant will take part in a total of 3 sets of experiment scenarios. These experiments may be spaced over a few days, and you will be asked to return the next day to continue with the user study. If you choose, the 3 sets can be conducted in one afternoon.

Instructions

In each experiment set, there will be 2 video viewing sessions. In each viewing session, you will view a sequence of video clips on a mobile device interface. Each video clip is shown one after the other, in a sequence. You are required to give a single **up or down** vote for each video that you view, based on the **assigned usage task** (see next section). The system may adjust and tailor the subsequent videos accordingly.

- If you like a video, vote it **up** (press the vote up button).
- If you dislike a video, vote it **down** (press the vote down button).
- You may interrupt and vote at any time during the viewing of each video clip
- Please provide a vote any time before the end of the video clip
- A vote has to be provided for every video clip
- Only 1 vote can be submitted per video clip
- If a video is voted up, the system will continue showing the current video (since you indicated you liked it)
- If a video is voted down, the system will proceed immediately to the next video.
- When 15 videos have been watched, a small window will display. Inform the test supervisor you have completed the viewing session.

Figure 27: User experiment instructions page 1

At the end of each viewing session, you will be provided with a short user survey. Finally, at the end of the experiment set, you will be provided with the user exit interview.

You will first be given 5 minutes to familarize yourself with the system and ask any questions you may have before the experiment begins.

Usage Task

For the purposes of this experiment set, assume you are interested in news stories about:

- North Korea, nuclear test, South Korea, related news items
- Middle East stories (Iraq, Afghanistan etc) and related news items

Important Notes

Some things that you should know about your participation:

- As you use the system, please do so as you would if using a real system on your mobile phone or portable device.
- You are encouraged to "think-aloud" and describe your actions and feelings to the test supervisor as you use the system.
- This is not a test of you; you are testing the system. Do not worry about making "mistakes".
- There is no right or wrong answer. We are interested to know if the system is suitable for users.
- If you ever feel that you are lost or do not understand what you have been given, please let the test supervisor know. I will clarify or explain further.
- We will be video recording this session for further study if needed. Your name will not be associated or reported with data or findings from this evaluation.

Do you have any questions before we begin?

Figure 28: User experiment instructions page 2

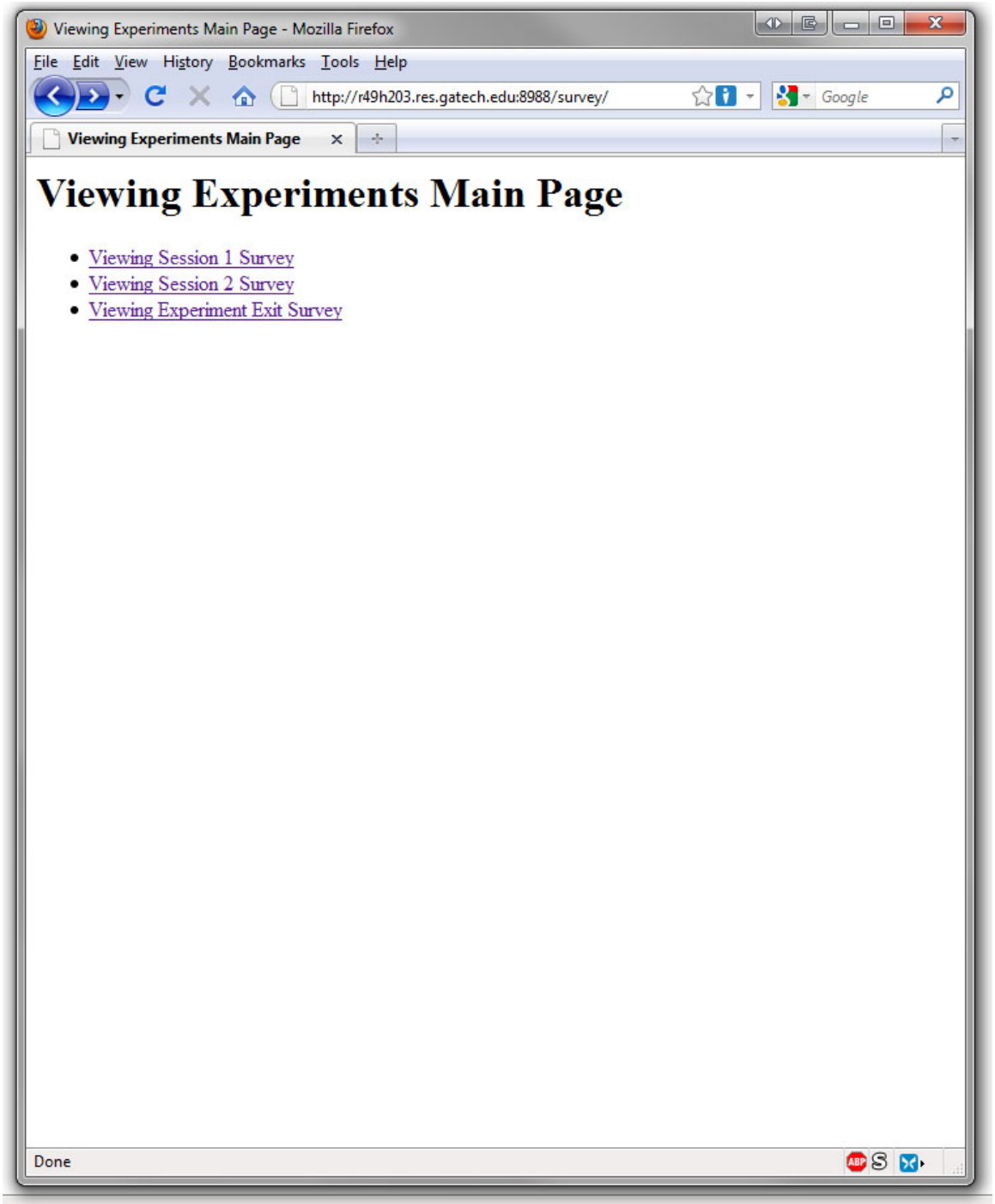


Figure 29: Online survey index page

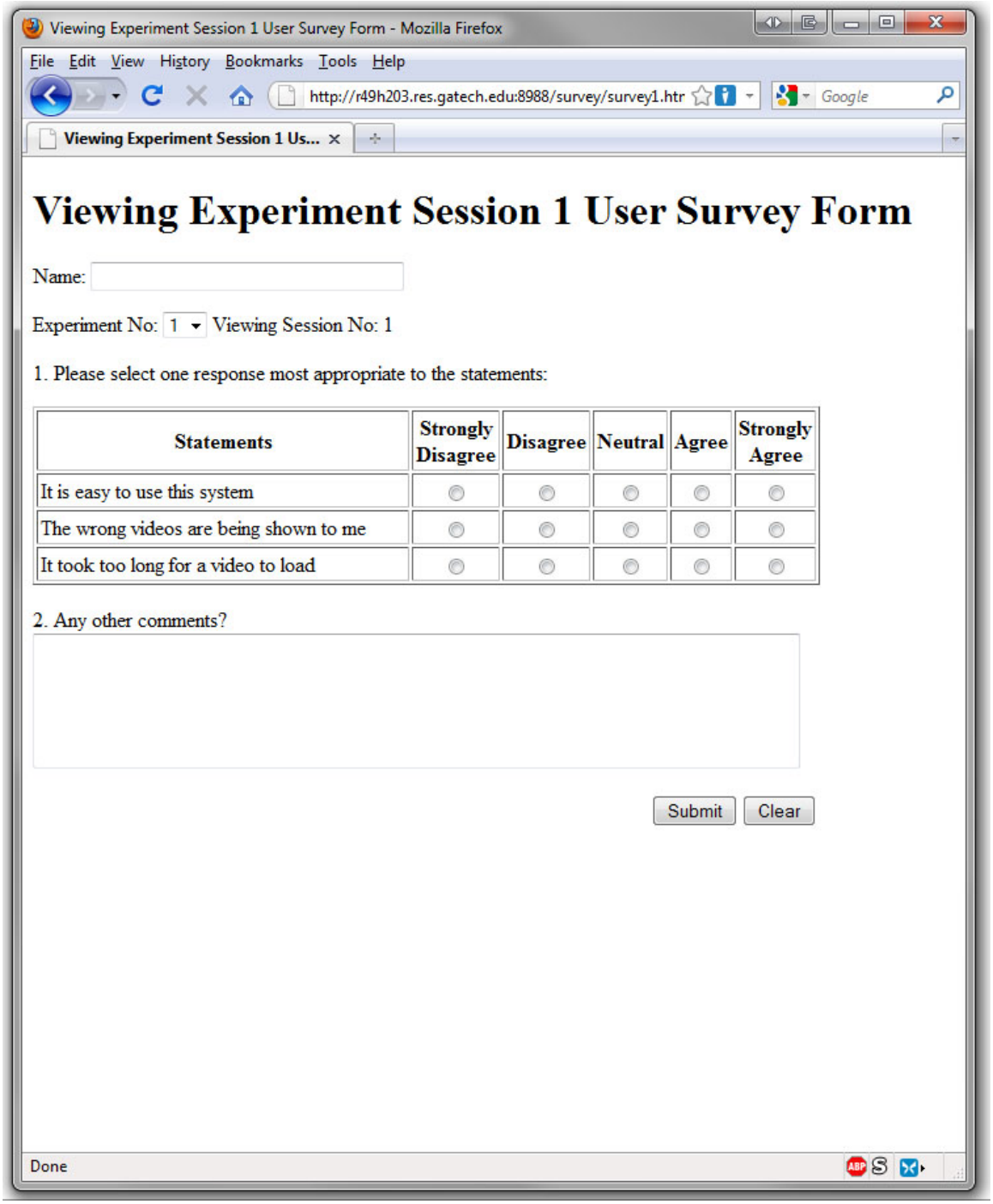


Figure 30: Session 1 online survey form

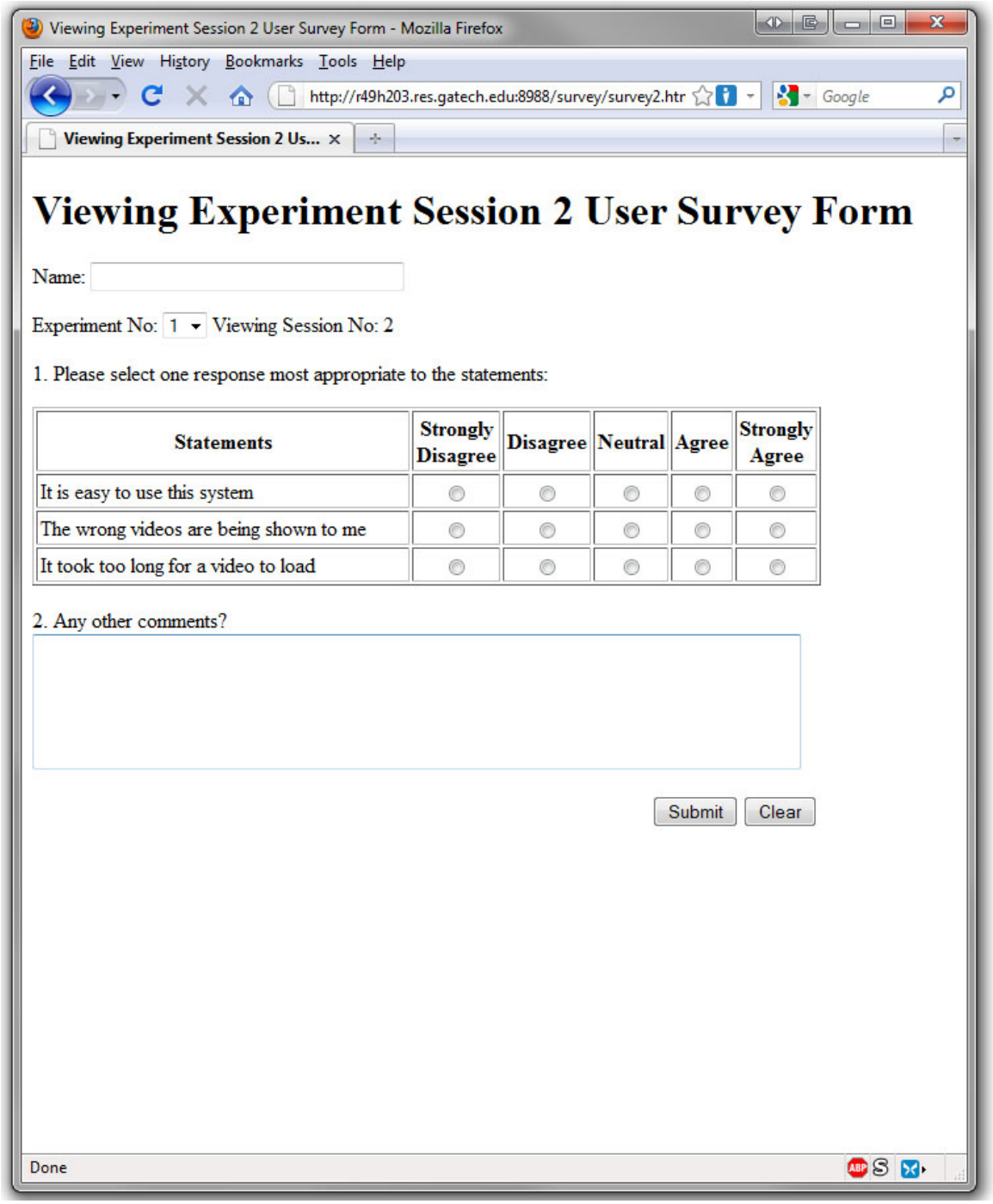


Figure 31: Session 2 online survey form

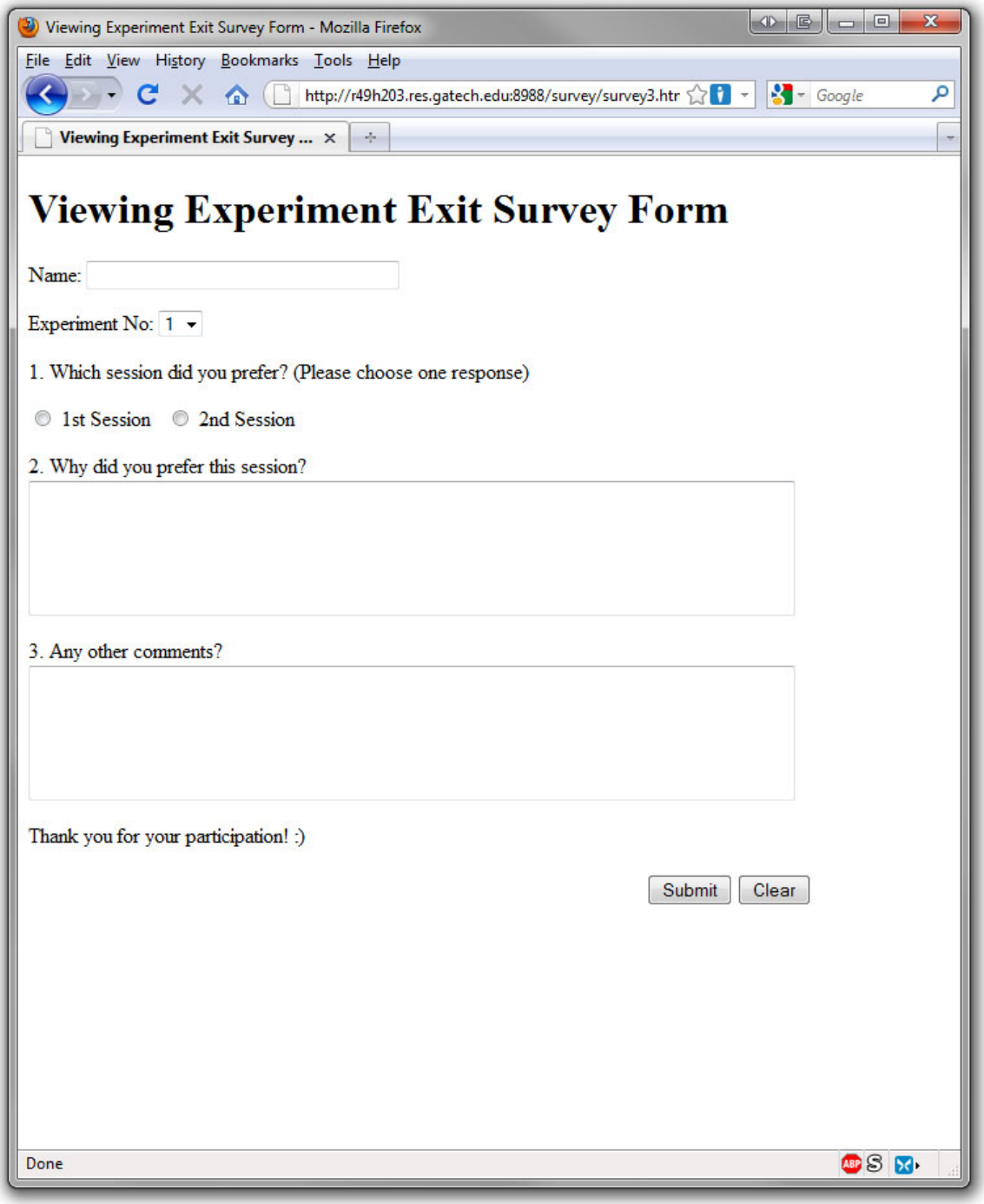


Figure 32: Online exit survey form

Viewing Session 1 User Survey Form

Name:

Experiment No.: Viewing Session No.: 1

1. Please select one response most appropriate to the statements:

Statements	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
It is easy to use this system					
The wrong videos are being shown to me					
It took too long for a video to load					

2. Any other comments? (Please write in the space below)

Figure 33: Session 1 paper-based survey form

Viewing Session 2 User Survey Form

Name:

Experiment No.: Viewing Session No.: 2

1. Please select one response most appropriate to the statements:

Statements	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
It is easy to use this system					
The wrong videos are being shown to me					
It took too long for a video to load					

2. Any other comments? (Please write in the space below)

Figure 34: Session 2 paper-based survey form

Viewing Experiment Exit Survey Form

Name:

Experiment No.:

Please answer the following 2 questions:

1. Which viewing session did you prefer? (Please circle one response)

1st Session 2nd Session

2. Why did you prefer this set? (Please write in the space below)

3. Any other comments? (Please write in the space below)

Thank you for your participation!

Figure 35: Paper-based exit survey form

APPENDIX C

USER EXPERIMENT SCREENSHOTS

Figure 36: Vertical orientation (HTC Dream)

Figure 37: Horizontal orientation (HTC Magic)

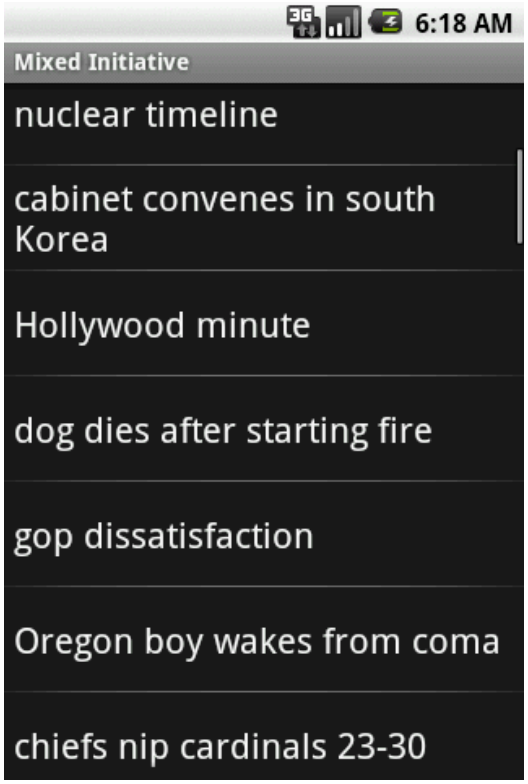


Figure 38: Manual video selection mode

Figure 39: Vote up dialog

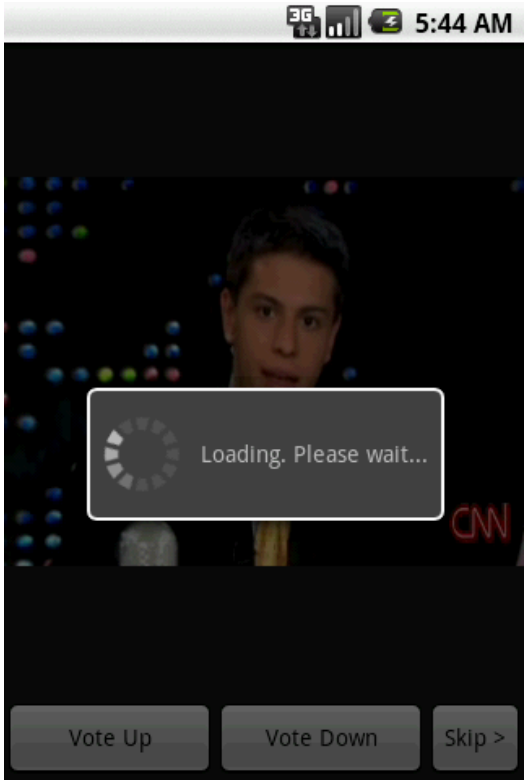


Figure 40: Video loading dialog

Figure 41: Settings menu for user experiment

APPENDIX D

USER EXPERIMENT SURVEY RESULTS

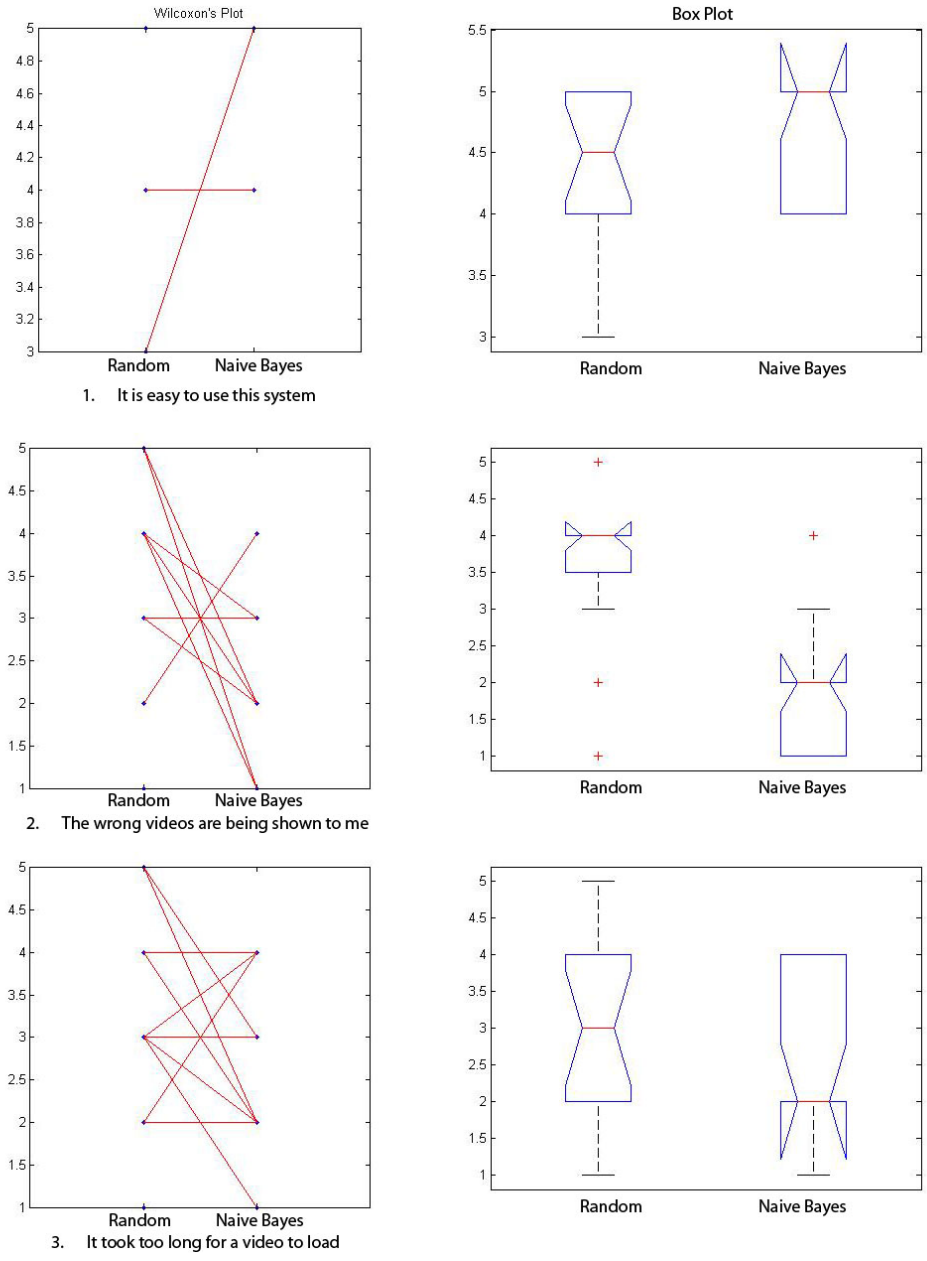


Figure 42: User experiment 1 survey plots

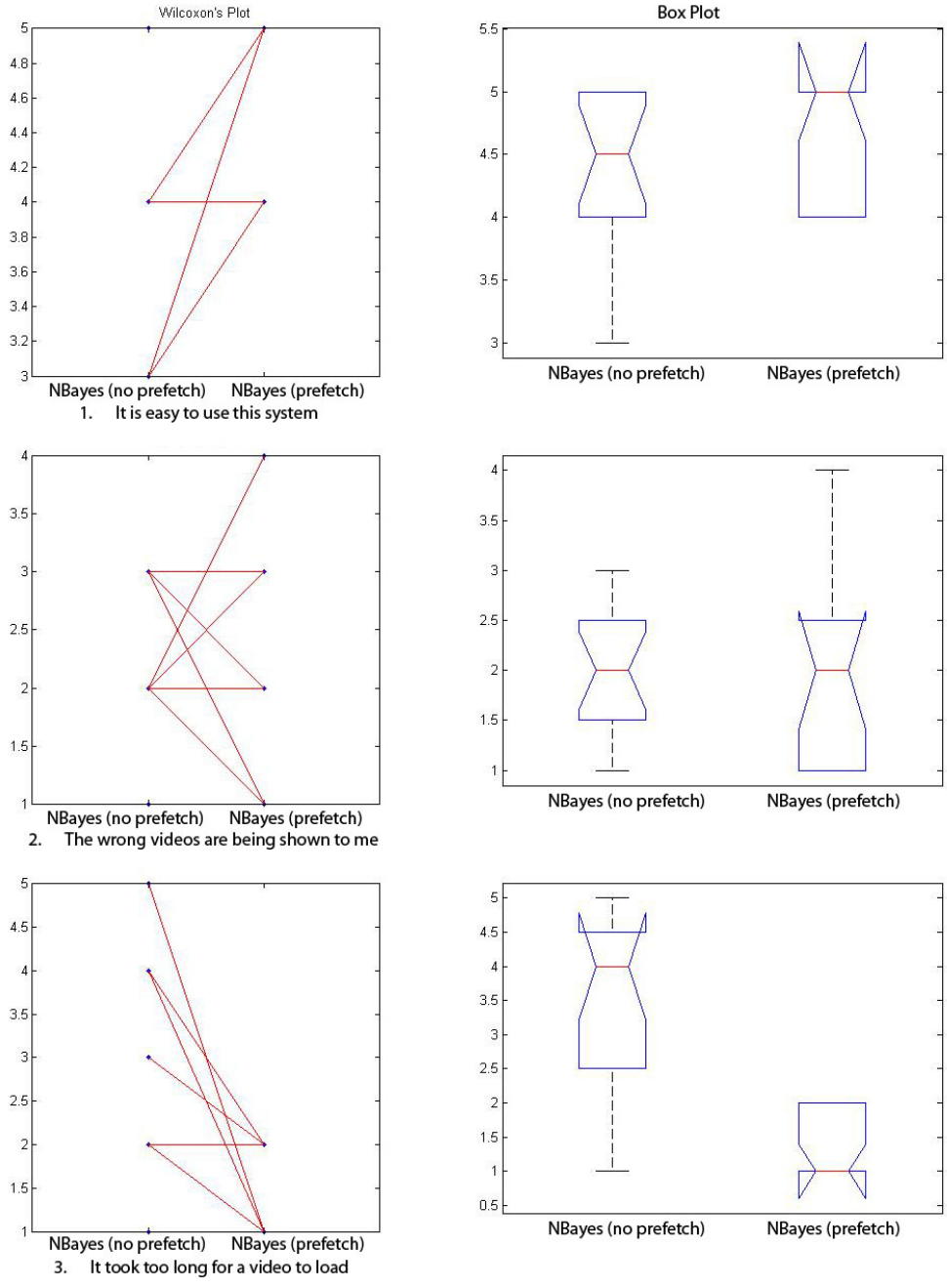


Figure 43: User experiment 2 survey plots

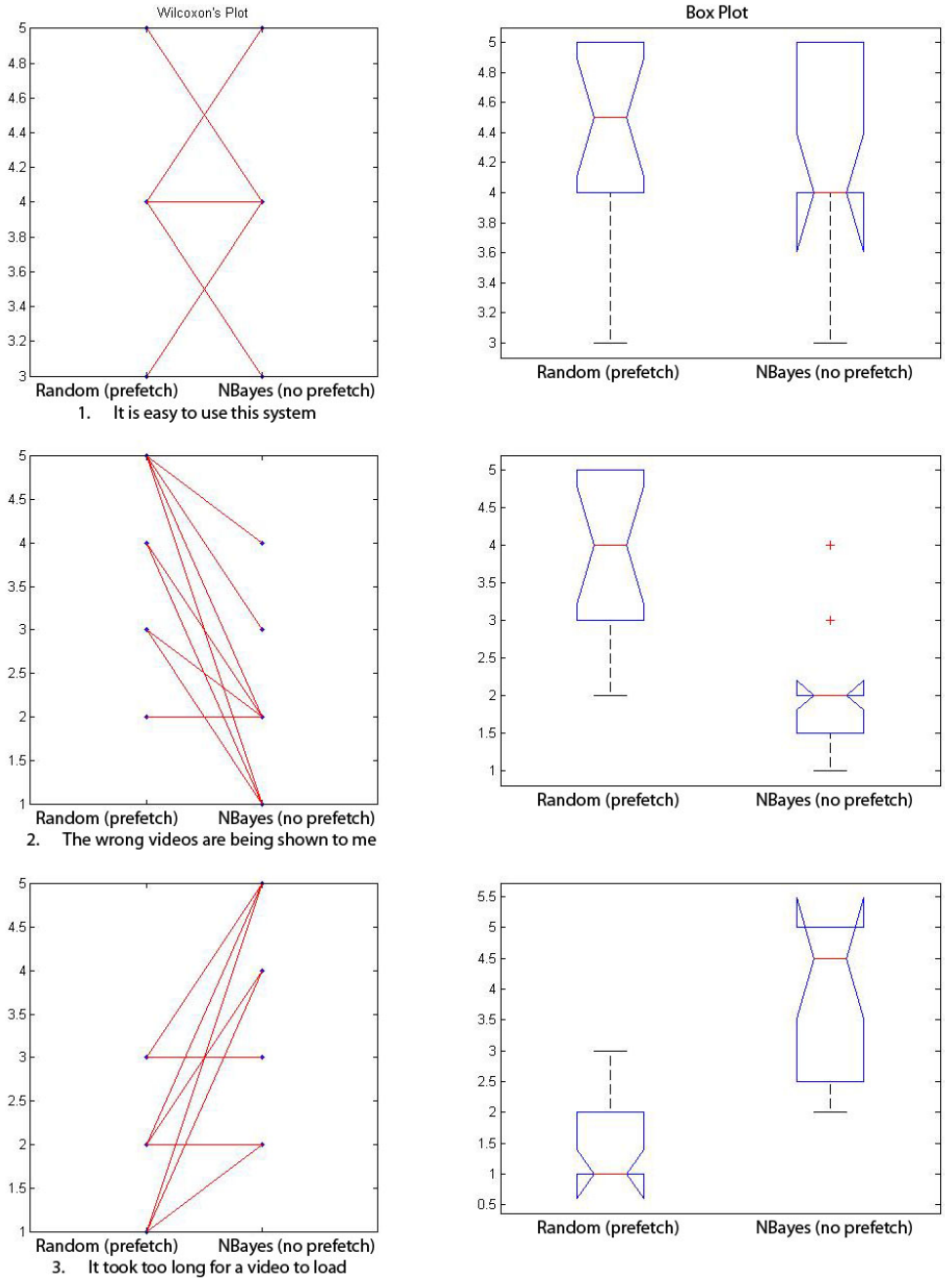


Figure 44: User experiment 3 survey plots

REFERENCES

[1] S. B. Newsam Shawn and B. S. Manjunath, "Category-based image retrieval," in *In Proc. IEEE International Conference on Image Processing, Special Session on Multimedia Indexing, Browsing and Retrieval*, pp. 596–599, 2001.

[2] C. D. Manning, P. Raghavan, and H. Schtze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[3] YouTube, "5 stars dominate youtube ratings." `http://youtube-global.blogspot.com/2009/09/five-stars-dominate-ratings.html`.

[4] M. Satyanarayanan, "Fundamental challenges in mobile computing," in *PODC '96: Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing*, (New York, NY, USA), pp. 1–7, ACM, 1996.

[5] H. Knoche and J. McCarthy, "Design requirements for mobile tv," in *Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices and Services MobileHCI 05*, pp. 69–76, MobileHCI05, 2005.

[6] D. A. Norman, *The Design of Everyday Things*. Basic Books, first ed., 2002.

[7] C. Gurrin, L. Brenna, D. Zagorodnov, H. Lee, A. F. Smeaton, and D. Johansen, "Supporting mobile access to digital video archives without user queries," in *MobileHCI '06: Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*, (New York, NY, USA), pp. 165–168, ACM, 2006.

[8] A. Marcus, J. Ferrante, T. Kinnunen, K. Kuutti, and E. Sparre, "Baby faces: user-interface design for small displays," in *Proceedings of the International ACM Conference on Computer-Human Interaction (CHI 98)*, 1998.

[9] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 2, no. 1, pp. 1–19, 2006. 1126005.

[10] J. Lahti, U. Westermann, M. Palola, J. Peltola, and E. Vildjiounaite, "Mobicon: integrated capture, annotation, and sharing of video clips with mobile phones," in *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, (New York, NY, USA), pp. 798–799, ACM, 2005.

[11] S. Ahern, S. King, and M. Davis, "Mmm2: mobile media metadata for photo sharing," in *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, (New York, NY, USA), pp. 790–791, ACM, 2005.

[12] E. Horvitz, "Principles of mixed-initiative user interfaces," in *Proceedings of the ACM SIGCHI Conf. Human Factors in Computing Systems*, 1999.

[13] J. E. Allen, C. I. Guinn, and E. Horvitz, "Mixed-initiative interaction," *Intelligent Systems and Their Applications, IEEE [see also IEEE Intelligent Systems]*, vol. 14, no. 5, pp. 14–23, 1999.

[14] J. S. A. Lee and N. Jayant, "**Mixed-initiative multimedia for mobile devices: a voting-based user interface for news videos**," in *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, (New York, NY, USA), pp. 611–614, ACM, 2006.

[15] J. S. A. Lee and N. Jayant, "**Gestures for mixed-initiative news video browsing on mobile devices**," in *MM '09: Proceedings of the seventeen ACM international conference on Multimedia*, (New York, NY, USA), pp. 1011–1012, ACM, 2009.

[16] J. S. A. Lee and N. Jayant, "**Mixed-initiative multimedia for mobile devices: design of a semantically-relevant low-latency system for news video recommendations**," in *Proceedings of IEEE SEC 2009*, March 2009.

[17] J. S. A. Lee and N. Jayant, "**Mixed-initiative news video recommendations and browsing for mobile devices**," in *ICSC '09: Proceedings of the IEEE International Conference on Semantic Computing*, pp. 569–570, 2009.

[18] Google, "Google search engine." `http://www.google.com`.

[19] Microsoft, "Bing." `http://www.bing.com`.

[20] Y. Rui, T. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: a power tool for interactive content based image retrieval," *Circuits and Systems for Video Technology, IEEE Transactions on*, 1998.

[21] G. Salton and C. Buckley, "Improving retrieval performance by relevance feedback," *Journal of the American Society for Information Science*, 1990.

[22] S. Santini, A. Gupta, and R. Jain, "Emergent semantics through interaction in image databases," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 13, no. 3, pp. 337–351, 2001.

[23] G. Salton and M. McGill, *Introduction to modern information retrieval*. New York, NY: McGraw Hill, 1983.

[24] G. Salton, *The SMART Retrieval System - Experiments in Automatic Document Processing*. Englewood Cliffs, NJ: Prentice Hall, 1971.

[25] J. Rocchio, "Relevance feedback in information retrieval," in *The SMART Retrieval System*, pp. 313–323, Englewood Cliffs, NJ, 1971.

[26] X. S. Zhou and T. S. Huang, "Exploring the nature and variants of relevance feedback," in *CBAIVL '01: Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries*, (Washington, DC, USA), p. 94, IEEE Computer Society, 2001.

[27] YouTube, "Youtube." `http://www.youtube.com`.

[28] D. Kelly and J. Teevan, "Implicit feedback for inferring user preference: a bibliography," in *ACM SIGIR Forum*, 2003.

[29] M. Claypool, P. Le, M. Wased, and D. Brown, "Implicit interest indicators," in *Proceedings of the 6th international conference on Intelligent user interfaces*, 2001.

[30] M. Claypool, D. Brown, P. Le, and M. Waseda, "Inferring user interest," *Internet Computing, IEEE*, 2001.

[31] D. Oard and J. Kim, "Implicit feedback for recommender systems," in *Proceedings of the AAAI Workshop on Recommender Systems*, 1998.

[32] Y. Hijikata, "Implicit user profiling for on demand relevance feedback," in *Proceedings of the 9th international conference on Intelligent user intrfaces*, 2004.

[33] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, "Recommending and evaluating choices in a virtual community of use," in *ACM Conference on Human Factors in Computing Systems*, 1995.

[34] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "Grouplens: applying collaborative filtering to usenet news," *Commun. ACM*, vol. 40, no. 3, pp. 77–87, 1997.

[35] Amazon, "Amazon online bookstore." `http://www.amazon.com`.

[36] Netflix, "Netflix." `http://www.netflix.com`.

[37] Pandora, "Pandora internet radio." `http://www.pandora.com`.

[38] Last.fm, "Last.fm music service." `http://www.last.fm`.

[39] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: An open architecture for collaborative filtering of netnews," in *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pp. 347–350, 1994.

[40] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *ACM SIGIR conference on research and development in information retrieval*, pp. 253–260, ACM, 2002.

[41] R. J. Mooney and L. Roy, "Content-based book recommending using learning for text categorization," in *Proceedings of the fifth ACM conference on digital libraries*, pp. 195–204, ACM Press, 2000.

[42] J. G. Apostolopoulos, W. tian Tan, and S. J. Wee, "Video streaming: Concepts, algorithms, and systems," tech. rep., HP Laboratories, 2002.

[43] L. d'Haenens, N. Jankowski, and A. Heuvelman, "News in Online and Print Newspapers: Differences in Reader Consumption and Recall," *New Media Society*, vol. 6, no. 3, pp. 363–382, 2004.

[44] "A new model for news: Studying the deep structure of young-adult news consumption," research report, Associated Press Context Based Research Group, 2008.

[45] CNN, "CNN video." `http://www.cnn.com/video/`.

[46] M. Porter, "An algorithm for suffix stripping," in *Program*, vol. 14(3), pp. 130–137, 1980.

[47] D. Lewis, "Naive (bayes) at forty: The independence assumption in information retrieval," in *Proceedings of ECML-98, 10th European Conference on Machine Learning*, 1998.

[48] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," in *Machine Learning*, vol. 29, p. 103137, 1997.

[49] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, K. V. Ch, G. Paliouras, and C. D. Spyropoulos, "An evaluation of naive bayesian anti-spam filtering," in *Proceedings of the Workshop on Machine Learning in the New Information Age*, pp. 9–17, 2000.

[50] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Addison Wesley, 1999.

[51] D. Duchamp, "Prefetching hyperlinks," in *Proc. of the second USENIX Symposium on Internet Technologies and Systems*, 1999.

[52] V. Padmanabhan and J. C. Mogul, "Using predictive prefetching to improve world wide web latency," in *ACM SIGCOMM Computer Communication Review*, vol. 26, pp. 22–36, 1996.

[53] M. Crovella and P. Barford, "The network effects of prefetching," in *Proceedings of IEEE INFOCOM*, 1998.

[54] T. M. Kroeger, D. D. E. Long, and J. C. Mogul, "Exploring the bounds of web latency reduction from caching and prefetching," in *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, 1997.

[55] J. Yoon and N. Jayant, "Pre-fetching for content-based image retrieval," in *Proceedings of the International Conference on Multimedia and Expo ICME*, 2002.

[56] S. Sen, J. Rexford, and D. Towsley, "Proxy prefix caching for multimedia streams," in *Proceedings of IEEE INFOCOM*, 1999.

[57] S. Lee, W. Ma, and B. Shen, "An interactive video delivery and caching system using video summarization," in *Computer Communications*, vol. 25(4), pp. 424–435, Elsevier, 2002.

[58] S. Chen, B. Shen, S. Wee, and X. Zhang, "Adaptive and lazy segmentation based proxy caching for streaming media delivery," in *Proceedings of the Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, 2003.